A, C 30/3 93

(12)

DNA-TR-82-26

AD-A136822

FORCE BALANCE ALLOCATION MODEL

Stefan Shrier
System Planning Corporation
1500 Wilson Boulevard
Arlington, Virginia 22209

19 November 1982

Technical Report

CONTRACT No. DNA 001-82-C-0283

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

THIS WORK WAS SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE X384082169 Q85QAXOB00001 H2590D.

DTIC FILE COPY

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, DC 20305

Destroy this report when it is no longer needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY, ATTN: STTI, WASHINGTON, D.C. 20305, IF YOUR ADDRESS IS INCORRECT, IF YOU WISH TO BE DELETED FROM THE DISTRIBUTION LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DNA-TR-82-26	2. GOVT ACCESSION NO. AD-A136 822	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FORCE BALANCE ALLOCATION MODEL		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER 83-166S
7. AUTHOR(s) Stefan Shrier		8. CONTRACT OR GRANT NUMBER(s) DNA 001-82-C-0283
9. PERFORMING ORGANIZATION NAME AND ADDRESS System Planning Corporation 1500 Wilson Boulevard Arlington, Virginia 22209		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task Q85QAXOB-00001
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, DC 20305		12. REPORT DATE 19 November 1982
		13. NUMBER OF PAGES 65
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE N/A since UNCLASSIFIED

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This work was sponsored by the Defense Nuclear Agency under RDT&E RMSS Code Code X384082169 Q85QAXOB00001 H2590D.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Analytical Model
Damage Assessment
Weapon Allocator
Heuristic Decision Logic

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes an automated strategic force allocation model developed by the System Planning Corporation (SPC). The model was prepared to provide quick-to-use tools for evaluation and analysis. BASIC-coded programs, implemented on a TEKTRONIX 4051 computer and an IBM Personal Computer, calculate damage caused to targets (aggregated into hardness classes) by an arsenal of weapons. The arsenal is allocated against targets classes sequentially (i.e., in the order in which the classes appear in the database); class grouping (into "objectives")

20. ABSTRACT (Continued)

enables second weapon allocation to some classes to precede first weapon assignment to others.

The classes are characterized by number of targets within the class, target radius, and vulnerability index (VNTK). The classes are further characterized by time priority and type of damage expectancy sought. The arsenal consists of various systems specified by their number, explosive yield, probable error (CEP) accuracy, arrival probability, and type (missile or bomber).

Because of memory constraints, the allocator was designed to accommodate up to 20 weapon systems and 80 target classes within an objective. Auxiliary, peripheral storage on a disk (or tape) file is used to store the entire target database organized into objectives.

Typically, the model runs in minutes although the preparation time for arsenal files and target database is user dependent.

TABLE OF CONTENTS

LIST OF TABLES

FORCE BALANCE ALLOCATION MODEL

A. OVERVIEW

This report summarizes work sponsored by the Defense Nuclear Agency to provide the Studies, Analysis, and Gaming Agency (SAGA) with an improved stand-alone capability to perform force balance allocation analyses. System Planning Corporation (SPC) supplied this capability by developing computer software, implemented on a TEKTRONIX 4051, that calculates damage caused to targets in a database by an arsenal of weapons. The computer listing for the TEKTRONIX software is included as Appendix A to this report.

The software developed under this contract utilizes the Beginner's All-Purpose Symbolic Instruction Code (BASIC) computer programming language. This language was chosen by SAGA. Among its advantages, BASIC affords the user increased portability among microcomputers that are not dissimilar to the TEKTRONIX 4051.

The software design, development, and implementation effort produced a sequential allocator program that incorporates existing nonautomated procedures as well as existing probability-of-damage calculation (PD CALC) and damage expectancy calculation (DE CALC) programs. SAGA supplied the latter programs to SPC. SPC modified them to a BASIC subroutine format for implementation on the TEKTRONIX 4051.

Section B describes the target and weapon databases. Section C outlines the allocation methodology on which the computer programs are based. Section D discusses the algorithm and data structures. Section E describes user information. Appendices A and B contain computer listings for the programs designed and developed by SPC. Appendices C and D consist of materials provided by SAGA.

B. DATA DESCRIPTIONS

1. Target Database

The targets are aggregated into classes, each of which is characterized by the number of targets within the class, target radius, and hardness (VNTK). Additionally, goal attributes associated with each class describe the level and type of damage expectancy to be achieved. For the installation damage expectancy (IDE) goal type, the required damage level applies to each target in the class. For the mean installation damage expectancy (MIDE) goal type, the required damage level applies to the class as a whole.

The target classes are organized in five groups (called objectives): strategic nuclear threat, theater nuclear forces, leadership, conventional, and industry. Within each objective, target classes are further organized into priority groups to distinguish two priority types: time urgent and time critical targets.

2. Arsenal Database

The nuclear-weapon arsenal consists of various weapon systems specified by their number; explosive yield; accuracy, measured as circular error probable (CEP); arrival probability (both generated and day-to-day); and type (missile or bomber).

C. ALLOCATION METHODOLOGY DESCRIPTION

The methodology consists of two phases. In the first phase, a damage expectancy (DE) matrix is formed by computing the effect of each type of weapon on every target class in the data base. In phase two, the weapons are allocated against targets.

The nuclear arsenal is allocated against targets sequentially by objectives according to procedures developed at SAGA. An example of one procedure is described in a SAGA report included as Appendix D.

The procedure incorporates various decision and selection rules. Some of these rules are explicit, such as a weapon-selection priority rule, whereas some are implicit. For example, an implicit rule is imposed on the allocation scheme by ordering within the target database (e.g., leadership targets precede the industry targets). As an example of an explicit rule, first weapons within an objective group are allocated to all classes, and a second weapon is then applied where necessary to resolve goal shortfalls (if any occur) before moving on to the next objective. Figure 1 represents the allocation system flow diagram.

D. SEQUENTIAL ALGORITHM LOGIC

1. Program Overview

The BASIC computer programs, implemented on SAGA's TEKTRONIX computer, comprise three modules: the DE subroutines ("DAMAGE"), the allocator ("ALLOCATE"), and the driver module. TEKTRONIX computer listings appear in Appendix A. Constraints imposed by the TEKTRONIX BASIC interpreter make line-by-line annotation, mnemonic-name selection, and structured-control constructs (such as IF...THEN...ELSE) difficult. Moreover, to conform to TEKTRONIX memory and execution-time constraints and also to provide the model as much capacity as possible (i.e., number of target classes per objective and weapon types), the internal annotation was held to a minimum.

The programs that SPC developed follow the SAGA methodology, described in Appendix D, as modified by several extensive reviews of preliminary designs and flow diagrams with SAGA. The logic agreed upon follows the descriptions in Appendix D with the following exceptions:

- Strict compliance to DE goals is required.
- No more than two weapons per target are allocated.
- The number of second weapons required to meet MIDE shortfalls is computed by an exact, integer solution instead of the continuous, logarithmic approximation.

Pursuant to direction by SAGA personnel, the term "time critical" is used in lieu of "non-time urgent" found in Appendix D.

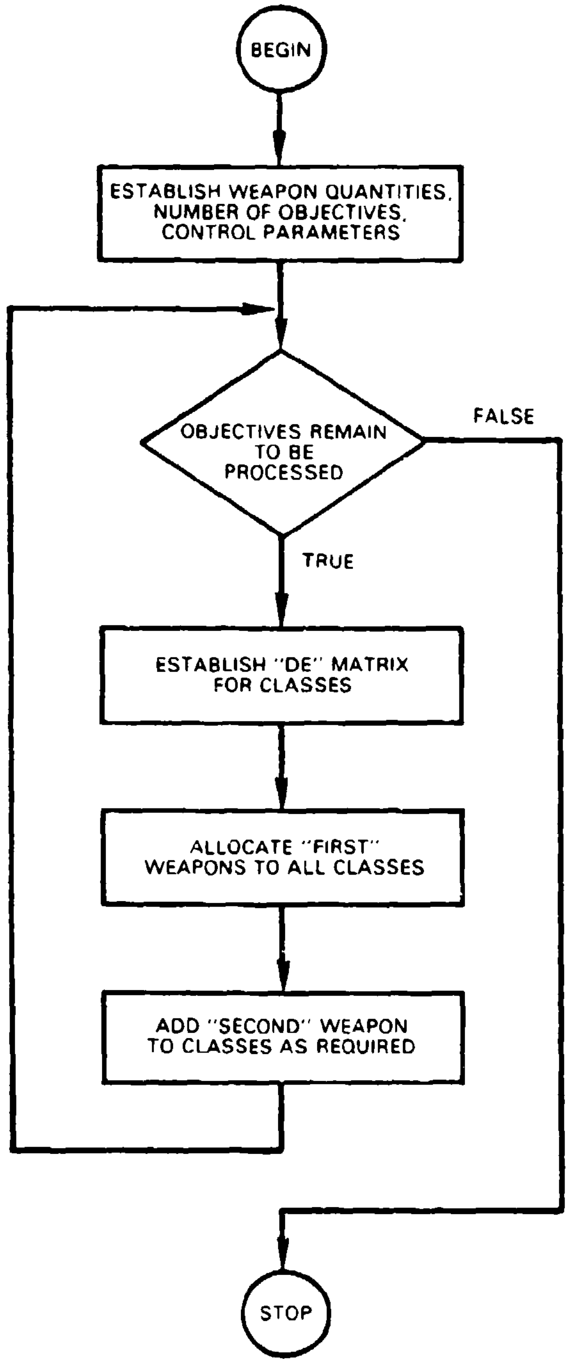

FIGURE 1.
ALLOCATION SYSTEM FLOW DIAGRAM

Fully documented versions of the BASIC programs appear in Appendix B. These programs, prepared and run on an IBM Personal Computer, played dual roles in the design and development of the TEKTRONIX programs: they not only validated the decision logic but also served as the "program design language" for the force balance model.

The following sections describe the data and algorithm structures used to implement the model and include programmer's notes.

2. Damage Expectancy Calculation

Weapon system DE against a target, defined as the product of arrival probability and destruction probability, is computed during the first phase of the methodology described earlier. The computer program, called DAMAGE, was prepared by consolidating two existing programs--listed in Appendix C--and modifying them to produce and format data needed in the second phase. Although no documentation, apart from these computer listings, was available to explain the algorithms used, they appear to conform to other programs in use at SAGA.

DAMAGE was validated to compute the same results as PD CALC and DE CALC by comparing them to various test cases. No algorithms in the original were altered or modified in any substantive way; however, portions of structurally "dead" code (such as BASIC lines 3200-3290 in DE CALC) and unreferenced variables were identified and removed.

The main component in DAMAGE initializes data tables, reads weapon-system and target data from data files, and prepares the DE matrix. Seven subroutines that support the computations perform the functions listed in Table 1. The BASIC line numbers are provided as a cross-reference to the TEKTRONIX listing in Appendix A.

TABLE 1. DAMAGE SUBROUTINES

Function	BASIC Line Numbers (Appendix A)
Interpret VNTK input	1510-1625
Prepare and write DE matrix	1630-1745
Compute DE	1750-1855
Compute weapon radius	1860-1930
Compute optimum burst height (air-burst assumption)	1935-1965
Compute destruction probability	1975-2055
Sort DE matrix entries	2060-2110

3. Data Structure Descriptions

Because of TEKTRONIX memory constraints, the force balance allocator was designed to accommodate up to 20 weapon systems and 80 target classes within each objective. In addition, to conserve computer memory and accommodate output-table requirements, several packed data structures were used in the model. Communication between the two automated phases (and consequently, modules DAMAGE and ALLOCATE) occurs through data files. The data files will be described later.

Important variables in the allocation model are:

T (,) This 80-target by 20-weapon DE table contains DEs ordered within each row into two halves and sorted within each half by increasing DE. The left half refers to the missile weapons, the right half to bomber weapons. In phase one, DAMAGE computes the DE of every weapon-type applied to each target CLASS. The weapon-type index and the DE against target CLASS is encoded into row CLASS in array T(,) in the form "vx.yz," where "vx" is the two-digit-integer weapon index and ".yz" is the damage expectancy. After first weapon allocation in phase two, T(,) encodes the number of weapons of given type applied as "stuvx.yz," where "stu" is the number of weapons applied of type "vx" with DE of ".yz."

W ( )	This 20-element array describes the number of weapons (remaining) in the arsenal.
G ( )	This 80-element packed array describes target goals. The fractional two-digit part is the moderate DE-level sought. The integer four-digit part encodes goal type as the high order digit (critical MIDE as 1, critical IDE as 2, time urgent MIDE as 3, and time urgent IDE as 4). The second highest digit is 0, 1, or 2, depending on whether coverage or moderate or high goals are sought. The remaining two digits are the high DE-level sought, expressed as a percent.
D ( )	This 80-element array represents the total goal deficit for each class. For an MIDE class, DE "points" are defined as the product of the DE level sought and the number of targets in the class. Then for MIDE targets, D ( ) is the number of DE points that remain to be achieved, expressed as a negative number. For IDE targets, D ( ) is the number of targets that have not been damaged to the specified level, also expressed as a negative number.
A ( )	This array implements a data stack (with stack-pointer A0) of packed, second-weapon allocations. The entries are of the form "stuvx.yz." The three high-order digits "stu" represent the number of type "vx" weapons compounded with first weapon type "yz" applied to a target class.
N ( )	This 80-element array contains the number of targets in any class.

4. Allocation Program Logic

The ALLOCATE programs consist of 15 subroutines with functions that are listed in Table 2. The BASIC line numbers refer to the TEKTRONIX computer listing in Appendix A. The names are provided to guide cross-referencing to the annotated listing in Appendix B. The annotated listing describes the decision logic; for example, "ALLOC1" describes first weapon selection and "ADD" describes second-weapon selection scanning.

For the first weapon, ALLOC1 allocates the lowest DE weapon that meets the goal. If no available weapon meets the goal, the highest DE weapon available is allocated. Time urgent targets get missile weapons, whereas time critical targets get bomber weapons.

ADD selects bomber weapons as the second weapon. For MIDE targets, it selects the highest DE weapons available. Otherwise, it selects the lowest DE weapon which, when compounded with the first weapon, meets the DE goal. If such a second weapon is not found, the highest DE weapon available is selected.

The allocation program logic incorporates several features that were suggested by SAGA personnel after review of earlier versions of the program. It also has many features designed to provide flexibility to accommodate change. The decomposition into phases and program block structure allows the user to insert additional decision rules; for example, after all time urgent target requirements are met, missile weapons could be used as second weapons.

Besides goal attainment, a coverage methodology could be employed that would allocate one weapon for each target, selecting the highest DE weapon remaining in the arsenal to be applied next.

E. USER INFORMATION

This section describes file organization and program execution. Program execution is conducted via TEKTRONIX BASIC drivers that access three user-created files of BASIC DATA statements saved as disk files. Phase one requires two user-prepared files: ARSENAL, describing weapon-systems, and TGT.DB, describing the target database. Phase two requires the file INVENTORY, which provides the quantities of each weapon type available for allocation. Communication between the two automated phases (i.e., passing the DE matrix) occurs via file DE.TBLE created by DAMAGE in phase one and accessed by ALLOCATE during phase two. DE.TBLE records the data structure T (,) described in the previous section. The BASIC drivers (called "PHASE1" and "PHASE2") use the direct commands APPEND and DELETE.

The number of classes cannot be greater than 80 (per objective), and the number of weapon-systems cannot exceed 20. If more computer memory is available, this restriction can be removed.

TABLE 2. ALLOCATE SUBROUTINES

Name*	Function	BASIC line numbers (Appendix A)
	Data declaration and program control	1000-1115
	Input weapon inventory	1120-1205
	Input DEs for classes within an objective	1210-1265
ALLOC1	Allocate first weapon	1270-1345
DEGOAL	Unpack goal levels and types	1350-1440
SCAN(r)	Scan right in DE table (first weapon)	1445-1580
SCAN(l)	Scan left in DE table (first weapon)	1585-1695
HOWMNY	Determine how many weapons of given DE fill need	1700-1750
APPLY	Apply first weapon	1755-1830
ALLOC2	Determine if shortfall exists (needs second weapon)	1835-1925
ADD	Try to attain goal by second weapon	1930-2065
SCANLF	Scan left in DE table (second weapon)	2070-2175
APPLY2	Apply second weapon	2180-2255
SCANRT	Scan right in DE table (second weapon)	2260-2390
REPORT	Display results	2395-2525

*Cross-reference to annotated listing.

1. Target Database

BASIC DATA statements beginning at line 4000, and saved as disk file TGT.DB, contain the following information:

- Number of objectives
- Number of classes in first objective
- First class: descriptor (six-character literal)
  goal type (i.e., type 1, 2, 3, or 4)
  VNTK (four-character literal)
  radius (in feet)
  number of targets
  high-DE goal
  moderate-DE goal
- Second class: descriptor
  goal type
  VNTK
  radius
  number of targets
  high-DE goal
  moderate-DE goal

  .
  .
  .

- Last class (in first objective)
- Number of classes in second objective (followed by second objective classes as shown above)

  .
  .
  .

- Last class (in last objective).

An example of the file setup would be:

```
4000 DATA 1
4005 DATA 4
4010 DATA "TGT1",4,"40P2",100,25,0.9,0.5
4015 DATA "TGT2",4,"38P5",100,73,0.5,0.5
4020 DATA "TGT3",3,"51L4",100,300,0.7,0.5
4025 DATA "TGT4",3,"48P7",100,653,0.8,0.8
```

As noted earlier, the goal type is encoded as an integer: critical MIDE as 1, critical IDE as 2, time urgent MIDE as 3, and time urgent IDE as 4.

In addition, target hardness (vulnerability index VNTK) has the following restrictions:

- The target type must be one of: L, M, N, O, P, Q, R, S, T, U, X, or D, and
- VN cannot exceed four for X-type targets.

2. Arsenal Database

BASIC statements beginning at line 3000 (up to 3999), and saved as disk file ARSENAL, contain the following:

- Number of missile weapon types
- First weapon: descriptor (six-character literal)
yield (in kilotons)
CEP (in feet)
generated-case arrival probability
day-to-day arrival probability
- Second weapon: descriptor
yield
CEP
generated-case arrival probability
day-to-day arrival probability

.
.
.

- Last missile weapon
- Number of bomber weapon types (followed by weapon parameters as shown above

.
.
.

- Last bomber weapon.

An example of this file structure would be:

```
3000 DATA 5
3005 DATA "ic1",200,750,0.88,0.88
3010 DATA "ic2",325,500,0.86,0.86
3015 DATA "sl1",50,1000,0.81,0.81
3020 DATA "sl2",100,900,0.84,0.84
3025 DATA "sl3",350,500,0.88,0.88
3030 DATA 4
3035 DATA "bm1",1000,500,0.58,0.55
3040 DATA "bm2",250,1000,0.62,0.51
3045 DATA "bm3",750,500,0.8,0.75
3050 DATA "bm4",200,385,0.72,0.65
```

3. Inventory Database

BASIC statements beginning at line 3000, and saved as disk file INVENTORY, contain the number of missile weapons, the quantities of each missile weapon, the number of bomber weapons, and the quantities of each.

The following example shows the quantities of nine weapon systems, five missile and four bomber types.

```
3000 DATA 5
3005 DATA 1000,1000,1500,1000,500
3010 DATA 4
3015 DATA 1000,1000,1000,1000
```

4. Program Execution

After the data files are created and saved on the disk, the automated first phase is executed by running the BASIC driver "PHASE1." This program produces the DE matrix as screen output.

When phase one is completed, the second phase is executed by running the BASIC driver "PHASE2." The program will call for user input to select computation for "generated" or "day-to-day" cases and for damage goal levels moderate, high, or coverage. (The program currently computes either moderate or high goals specified. The coverage methodology is not implemented. If coverage is specified, the program defaults to moderate level goals.)

The program produces weapon allocations for each target class (by objectives) as screen output by displaying the modified DE matrix and the second weapon allocation stack described earlier. The output format (as well as the data structures described earlier) is modeled after existing forms SAGA uses for manual sequential allocation.

Table 3 shows sample output for 25 IDE targets, using the data from the previous section, after two rounds (i.e., the first and the second weapon allocation), trying to achieve a DE of 0.90.

TABLE 3. IDE SAMPLE OUTPUT

First Round			Second Round		
tu-ide	25	-25	tu-ide	25	0
	3.15			3.15	
	4.28			4.28	
	1.52			1.52	
	2.79			2.79	
	2505.81			2505.81	
	7.28			7.28	
	6.57			6.57	
	9.68			9.68	
	8.78			8.78	
				2508.05	

The first round shows 25 time urgent IDE targets and a shortfall (shown by -25) in meeting the DE goal. DE array elements (i.e., entries in T (,)) corresponding to this class show the missile DEs applied to this class followed by the bomber DEs in the packed format described earlier. In particular, missile system three attains a DE of 0.15, four attains 0.28, one attains 0.52, two attains 0.79, and five--the system applied to each target in the class--attains 0.81 per weapon on this target. Because the goal (i.e., 0.90) is not met on any target, the shortfall on all targets is indicated. The bomber weapons applied to this target class achieve 0.28 from system seven, 0.57 from system six, 0.68 from system nine, and 0.78 from system eight.

The second round shows 25 targets with "0" indicating that no shortfall remains. The DE matrix entries are repeated followed by second weapon

allocation stack (the array A ( ) described earlier). The entry "2508.05" indicates that 25 system eight weapons are compounded with 25 system five weapons, which were applied in round one.

Table 4 illustrates output for the 653 MIDE targets against which 0.80 DE is sought. The fir round shows that 478 system two weapons (at 0.67 each) and 175 system five weapons (at 0.70 each) were applied against the target class. The shortfall, expressed by the "-80," is resolved after the second round by compounding 327 system eight and system two weapons. These compound weapons, 151 type two and 175 type five weapons, achieve the MIDE goal for this target class.

TABLE 4. MIDE SAMPLE OUTPUT

First round			Second round		
tu-mide	653	-80	tu-mide	653	0
	3.07			3.07	
	4.15			4.15	
	1.35			1.35	
	47802.67			47802.67	
	17505.70			17505.70	
	7.18			7.18	
	6.55			6.55	
	9.59			9.59	
	8.74			8.74	
				32708.02	

Appendix A

TEKTRONIX COMPUTER ALLOCATOR MODEL LISTINGS

"PHASE1" DE Driver

```
100 PRINT 'L ALLOCATE'
105 PRINT 'PHASE 1'
106 M=MEMORY
110 APPEND 'DAMAGE';1000,5
115 APPEND 'ARSENAL';3000,5
120 APPEND 'TGT.DB';4000,5
121 RESTORE
125 GOSUB 1000
126 PRINT 'PHASE 1 COMPLETE'
127 RESTORE
135 DELETE 1005,6000
136 DELETE B,D,E,F,G,Q,B$,C$,L$,T$,W$,X$,Z$
137 DELETE I,J,C7,J1,M,Z,N,G1,G2,C,O,P,T,V,S,K,J3,L0,L1,H0
138 DELETE I4,I8,Y3,Y4,V1,W,R1,P9,D9
1000 REM TARGET FOR APPENDS
3000 REM TARGET FOR APPENDS
4000 REM TARGET FOR APPENDS
```

"PHASE2" Allocation Driver

```
100 APPEND 'allocate';1000,5
110 APPEND 'inventory';3000,5
120 PRINT 'begin phase 2'
130 GOSUB 1000
140 PRINT 'phase 2 complete'
1000 REM target for append
3000 REM target for append
```

"DAMAGE"

```
1000 REM ADAPTATION OF DECALC AND FD CALC ...............AUGUST 1982
1005 REM
1010 REM prepared by dr. stefan shrier
1015 REM system planning corp., 1500 wilson blvd.
1020 REM arlington, va. 22209, telephone 703-841-3621
1025 REM PREPARED FOR JCS/SAGA/SFD FOR USE IN FORCE BALANCE ALLOCATION
1030 REM
1035 DIM F(7,4),B(4,18),E(4,2)
1040 DIM D(5,20),Q(2)
1045 DIM G(6,5)
1050 REM STRING VARIABLE SIZES...
1055 DIM B$(4),L$(12),Z$(1),X$(2),C$(1),A$(300),T$(6),W$(6)
1060 REM SET CONSTANT TABLES...THROUGH LINE 1120
1065 Q(1)=1.04167
1070 Q(2)=1.0989
1075 L$='LPMNORSQTUXD'
1080 REM
1085 FOR J=1 TO 5
1090 FOR I=1 TO 6
1095 READ G(I,J)
1100 NEXT I
1105 NEXT J
1110 DATA 1.8622,-3.237,2.0771,-1.4128,0.40348,-0.045266
1115 DATA 1.843,-3.0865,1.7386,-1.0635,0.28357,-0.028466
1120 DATA 1.8095,-2.858,1.2935,-0.6232,0.13978,-0.011672
1125 DATA 1.7671,-2.636,1.006,-0.35646,0.06215,-0.0040752
1130 DATA 1.6984,-2.3264,0.74818,-0.18783,0.023284,-0.0010853
1135 REM
1140 FOR J=1 TO 4
1145 FOR I=1 TO 7
1150 READ F(I,J)
1155 NEXT I
1160 NEXT J
1165 DATA 8.214,-0.1118,5.265E-4,2.162E-5,-6.638E-7,7.132E-9
1170 DATA -3.064E-11,8.315,-0.1033,-7.908E-4,-9.039E-5,1.458E-5
1175 DATA -5.22E-7,5.726E-9,8.783,-0.1355,0.002355,-2.086E-4
1180 DATA 9.901E-6,-1.872E-7,1.227E-9,8.789,-0.112,-6.658E-5
1185 DATA -5.803E-4,5.853E-5,-1.905E-6,2.056E-8
1190 REM
1195 FOR I=1 TO 4
1200 FOR J=1 TO 18
1205 READ B(I,J)
1210 NEXT J
1215 NEXT I
1220 DATA -0.5779,-1.224,-1.957,-2.805,-3.812,-5.051,-6.658,-8.926
1225 DATA -12.73,-0.2889,-0.6119,-0.9782,-1.401,-1.901,-2.509,-3.284
1230 DATA -4.343,-6.01,1.569,3.33,5.352,7.742,10.72,14.63,20.2,28.96
1235 DATA 45,0.7989,1.729,2.837,4.196,5.915,8.179,11.32,15.05,24.43
1240 DATA 0.001376,-0.002069,-0.04773,-0.2253,-0.8418,-2.375,-5.982
1245 DATA -14.22,-35.56,-0.03991,-0.1869,-0.5116,-1.133,-2.23,-4.116
1250 DATA -7.356,-13.21,-25.37,-0.00631,-0.04695,-0.1545,-0.3457
1255 DATA -0.5135,-0.4769,0.3151,3.115,12.22,0.001152,0.009542
1260 DATA 0.5145,0.1812,0.4847,1.12,2.376,4.907,10.64
1265 REM
1270 FOR J=1 TO 2
1275 READ E(1,J),E(2,J),E(3,J),E(4,J)
1280 NEXT J
1285 DATA 7.171,-0.0251,-0.001886,2.278E-5
1290 DATA 7.154,-0.1576,0.005228,-6.618E-5
1295 REM
1300 REM UNKNOWN SOUGHT IS DAMAGE EXPECTANCY (I.E., F)
1305 REM CONSTANT FOR TGT DISTRIBUTIONS, USED IN FD CALCULATION
1310 REM (0.4 UNIFORM, 0.231 NORMAL)
1315 C7=0.231
```

```
1320 REM ***************************************************
1325 REM GET WEAPON DATA
1330 REM
1340 REM   NO. URGENT/YIELD,CEP,FA-GTD,FA-DTD,...
1345 READ U0
1350 FOR J=1 TO U0
1355 READ W$,D(1,J),D(2,J),D(3,J),D(4,J)
1360 NEXT J
1365 REM
1370 REM SAME FOR CRITICAL
1375 READ B0
1380 FOR J=1 TO B0
1385 J1=J+U0
1390 READ W$,D(1,J1),D(2,J1),D(3,J1),D(4,J1)
1395 NEXT J
1400 REM DELETE 3005,5050
1405 W0=U0+B0
1410 REM
1415 REM    MAIN PROCESS LOOP ************************************
1420 OPEN 'DE.TBLE'#3,'F',A$
1430 REM
1435 READ O1
1440 PRINT #3:O1
1441 PRINT O1
1445 FOR O=1 TO O1
1450 READ C1
1455 PRINT #3:C1
1456 PRINT C1
1460 FOR C=1 TO C1
1465 READ T$,M,B$,Z,N,G1,G2
1470 GOSUB 1515
1475 GOSUB 1750
1480 GOSUB 1630
1485 NEXT C
1490 NEXT O
1492 CLOSE 3
1495 RETURN
1500 REM
1505 REM
1510 REM SUBROUTINE TO INTERPRET VNTK INPUT
1515 Z$=SEG(B$,LEN(B$)-1,1)
1520 P=POS(L$,Z$,1)
1525 IF P THEN 1540
1530 PRINT 'TYPE MUST BE L,M,N,O,P,Q,R,S,T,U,X,D'
1535 STOP
1540 X$=SEG(B$,1,LEN(B$)-2)
1545 V=VAL(X$)
1550 IF P<6 OR P=12 AND V<57 OR (P<11 AND V<35) THEN 1570
1555 IF P=11 AND V>0 AND V<=4 THEN 1580
1560 PRINT 'VN IS OUT OF RANGE FOR TGT TYPE'
1565 STOP
1570 IF P<12 THEN 1580
1575 P=5
1580 S=0.1*P
1585 T=P
1590 IF P=11 THEN 1620
1595 IF P<6 THEN 1605
1600 S=S-0.5
1605 C$=SEG(B$,LEN(B$),1)
1610 K=VAL(C$)
1615 RETURN
1620 K=0
1625 RETURN
```

"DAMAGE" (cont.)

```
1630 REM SUBROUTINE TO PREPARE AND WRITE TABLE
1635 FOR J3=1 TO W0
1640 D(5,J3)=J3!D(5,J3)
1645 NEXT J3
1650 L0=1
1655 L1=U0
1660 GOSUB 2060
1665 L0=U0+1
1670 L1=U0+B0
1675 GOSUB 2060
1680 PRINT #3:T$
1681 PRINT #3:B$
1682 PRINT #3:N
1683 PRINT #3:M*100+INT(G1*100)+G2
1684 PRINT T$;B$;N;M*100+INT(G1*100)+G2
1715 FOR J3=1 TO W0
1720 PRINT #3: USING 1725:D(5,J3);
1721 PRINT USING 1725:D(5,J3);
1725 IMAGE 3d.2d
1730 NEXT J3
1745 RETURN
1750 REM SUBROUTINE TO COMPUTE DES
1755 H0=1
1760 J1=1
1765 I4=1
1770 IF T<6 THEN 1780
1775 I4=2
1780 I8=10*S
1785 REM          LOOP ON J3
1790 FOR J3=1 TO W0
1795 Y3=D(1,J3)^(1/3)
1800 IF T=11 THEN 1815
1805 GOSUB 1860
1810 GOSUB 1935
1815 J4=J3
1820 J5=1
1825 GOSUB 1980
1830 D(5,J3)=P*D(3,J3)
1835 Z$=SEG(L$,T,1)
1840 REM PRINT @A: USING 1055:V,Z$,K,V1,D(1,J3),D(2,J4),Z
1845 REM PRINT @A: USING 450:H0,W,P,D(3,J3),D1
1850 NEXT J3
1855 RETURN
1860 REM SUBROUTINE TO COMPUTE WEAPON RADIUS
1865 Y4=1/Y3
1870 V1=V
1875 IF K=0 THEN 1890
1880 J=9*I4-9+K
1885 V1=V+B(1,J)+Y4*(B(2,J)+Y4*(B(3,J)+Y4*B(4,J)))
1890 J=I4+2
1895 W=F(7,J)
1900 FOR I=6 TO 1 STEP -1
1905 W=W*V1+F(I,J)
1910 NEXT I
1915 W=Y3*EXP(W)
1920 W=W/(Q(I4)*(1-S^2))
1925 RETURN
1930 REM
```

"DAMAGE" (cont.)

```
1935 REM SUBROUTINE TO COMPUTE OPTIMUM HOB
1940 IF I4=1 OR V1=>15 THEN 1955
1945 H0=1661*EXP(-0.06138*V1)
1950 GO TO 1960
1955 H0=EXP(E(1,I4)+V1*(E(2,I4)+V1*(E(3,I4)+V1*E(4,I4))))
1960 H0=Y3*H0
1965 RETURN
1970 REM
1975 REM SUBROUTINE TO COMPUTE P D
1980 IF D(2,J4) THEN 1995
1985 P=1
1990 RETURN
1995 R1=W/SQR(D(2,J4)^2+C7*/ 2)
2000 IF S=0.1 AND R1<3 THEN 1985
2005 IF S=0.2 AND R1<3.5 THEN 1985
2010 IF S=0.3 AND R1<4.5 THEN 1985
2015 IF S=0.4 AND R1<6 THEN 1985
2020 IF S=0.5 AND R1<8 THEN 1985
2025 IF R1>0.1 THEN 2040
2030 P=0
2035 RETURN
2040 F9=R1*(G(5,18)+R1*G(6,18))
2045 F9=G(1,18)+R1*(G(2,18)+R1*(G(3,18)+R1*(G(4,18)+F9)))
2050 P=EXP(-EXP(F9))
2055 RETURN
2060 REM subroutine to order in table
2065 IF L1-L0<0 THEN 2110
2070 FOR I=L0+1 TO L1
2075 FOR J=L0 TO I
2080 IF D(5,I)-INT(D(5,I))>D(5,J)-INT(D(5,J)) THEN 2100
2085 D9=D(5,J)
2090 D(5,J)=D(5,I)
2095 D(5,I)=D9
2100 NEXT J
2105 NEXT I
2110 RETURN
2115 REM PROGRAM FOR DE/FD CONCLUDED
2120 REM RETURN STATEMENT IN MAIN PROCESS LOOP
3000 REM TARGET FOR APPEND
4000 REM target for append
```

"ALLOCATE"

```
1000 REM allocation program.................september 1982
1005 REM ..................................
1010 REM prepared by Dr. Stefan Shrier
1015 REM System Planning Corporation, 1500 Wilson Blvd.
1020 REM Arlington, Va. 22209, (703)-841-3621
1025 REM..................................
1030 REM prepared for JCS/SAGA/SFD
1035 REM..................................
1040 DIM T(80,20),A(20),D(80),G(80),W(20),N(80)
1045 DIM A$(300),R$(1)
1050 PRINT 'end of declarations'
1055 GOSUB 1120
1060 FOR O=1 TO O1
1065 GOSUB 1210
1070 FOR C=1 TO C1
1075 GOSUB 1275
1080 GOSUB 2395
1085 NEXT C
1090 FOR C=1 TO C1
1095 GOSUB 1845
1100 GOSUB 2395
1105 NEXT C
1110 NEXT O
1115 RETURN
1120 REM subroutine to input toggles and weapon inventory
1125 PRINT 'std-0 or dtd-1 (std override)'
1130 INPUT R$
1135 PRINT 'desgal cov-0, mod-1, or hi-2'
1140 INPUT P1
1145 REM echo here
1150 READ U0
1155 FOR J=1 TO U0
1160 READ W(J)
1165 NEXT J
1170 READ B0
1175 FOR J=1 TO B0
1180 READ W(J+U0)
1185 NEXT J
1190 W0=U0+B0
1195 OPEN 'de.tble'#1,'r',A$
1200 INPUT #1:O1
1205 RETURN
1210 REM subroutine to input des for classes within objective
1215 INPUT #1:C1
1220 FOR C=1 TO C1
1225 INPUT #1:T$
1226 INPUT #1:B$
1230 INPUT #1:N(C)
1235 INPUT #1:G(C)
1240 FOR J=1 TO W0
1245 INPUT #1:T(C,J)
1250 NEXT J
1255 NEXT C
1260 CLOSE 1
1265 RETURN
```

```
1270 REM ...................................................
1275 REM
1280 REM subroutine alloc1--allocate first weapon
1285 A0=0
1290 GOSUB 1350
1295 REM
1300 F3=1
1305 IF F3=0 THEN 1340
1310 GOSUB 1445
1315 IF J0=0 THEN 1330
1320 GOSUB 1760
1325 GO TO 1335
1330 F3=0
1335 GO TO 1305
1340 REM while-end (1430)
1345 RETURN
1350 REM ...................................................
1355 REM subroutine degoal -- unpack goals and categories for
1360 REM categories (tu/crit and ide/mide) <-> flags f1 and f2
1365 G0=INT(INT(G(C))/100)
1370 F1=G0=1 OR G0=3
1375 REM flag for (M)ide set. now set urgent/critical flag f2
1380 F2=G0=>3
1385 G0=G(C)/100-G0
1390 REMswitch h1 for level goals was read in earlier
1395 IF H1=2 THEN 1410
1400 G0=G(C)-INT(G(C))
1405 GO TO 1415
1410 G0=INT(G0*100)/100
1415 N0=N(C)
1420 IF F1=0 THEN 1435
1425 D(C)=-N0*G0
1430 GO TO 1440
1435 D(C)=-N0
1440 RETURN
1445 REM ...................................................
1450 REM subroutine scan(r) -- for first weapon
1455 REM scanning to the right in the dg table. J0 marks table
1460 REM chosen; select first weapon that meets 'per wpn' de g
1465 J0=0
1470 L0=1
1475 L1=U0
1480 IF F2 THEN 1495
1485 L0=U0+1
1490 L1=U0+B0
1495 J=L0
1500 IF J>L1 OR J0>0 THEN 1565
1505 T1=T(C,J)
1510 G1=T1-INT(T1)
1515 IF G1<G0 THEN 1555
1520 J1=INT(100*(T1/100-INT(T1/100)))
1525 IF W(J1)=0 THEN 1550
1530 J0=J
1535 X=G1
1540 Y=N0
1545 GOSUB 1705
1550 REM
1555 J=J+1
1560 GO TO 1500
1565 REM while-end (1635)
1570 IF J0>0 THEN 1580
1575 GOSUB 1590
1580 RETURN
```

```
1585 REM ..........................................................
1590 REM subroutine scan(L) for first weapon
1595 REM swap 10 and 11
1600 LY=L0
1605 L0=L1
1610 L1=LY
1615 J=L0
1620 IF J=0 OR J0>0 THEN 1680
1625 T1=T(C,J)
1630 J1=INT(100*(T1/100-INT(T1/100)))
1635 IF W(J1)=0 THEN 1670
1640 J0=J
1645 G1=T1-INT(T1)
1650 X=G1
1655 Y=N0
1660 GOSUB 1705
1665 REM
1670 J=J-1
1675 GO TO 1620
1680 REM while-end (1740)
1685 REM
1690 RETURN
1695 REM ..........................................................
1700 REM
1705 REM
1710 REM subroutine howmny -- args x(wpn de) and y(wpn stock)
1715 IF F1=0 THEN 1730
1720 M=INT(ABS(D(C))/X+1)
1725 IF M<=Y THEN 1735
1730 M=Y
1735 IF M<=W(J1) THEN 1745
1740 M=W(J1)
1745 RETURN
1750 REM ..........................................................
1755 REM
1760 REM
1765 REM subroutine apply -- first weapon
1770 W(J1)=W(J1)-M
1775 N0=N0-M
1780 T(C,J0)=T(C,J0)+100*M
1785 REM f3 set true if more tgts remain in class
1790 F3=N0>0
1795 REM
1800 IF F1 THEN 1820
1805 IF G0>G1 THEN 1830
1810 D(C)=D(C)+M
1815 GO TO 1830
1820 D(C)=D(C)+M*X
1825 F3=D(C)<0 AND N0>0
1830 RETURN
```

"ALLOCATE" (cont.)

```
1835 REM .....................................................
1840 REM
1845 REM
1850 REM subroutine alloc2 -- shortfall to goal...needs 2nd wer
1855 A0=0
1860 IF D(C)=>0 THEN 1920
1865 D1=D(C)
1870 GOSUB 1350
1875 REM
1880 D(C)=D1
1885 L0=1
1890 L1=U0
1895 IF F2 THEN 1910
1900 L0=U0+1
1905 L1=U0+B0
1910 GOSUB 1935
1915 REM
1920 RETURN
1925 REM .....................................................
1930 REM
1935 REM
1940 REM subroutine add -- try to achieve DE goals
1945 K=L0
1950 F3=1
1955 IF K>L1 THEN 2060
1960 T1=T(C,K)
1965 G1=T1-INT(T1)
1970 K0=K
1975 K1=INT(100*(T1/100-INT(T1/100)))
1980 IF F1 THEN 1990
1985 F3=G1<G0
1990 N1=INT(INT(T1)/100)
1995 IF N1=0 OR F3=0 THEN 2045
2000 IF F1 THEN 2015
2005 GOSUB 2270
2010 GO TO 2020
2015 GOSUB 2080
2020 IF J0>0 THEN 2035
2025 F3=0
2030 GO TO 2045
2035 GOSUB 2175
2040 GO TO 1995
2045 REM while-end (3155)
2050 K=K+1
2055 GO TO 1955
2060 REM while-end (3120)
2065 RETURN
```

"ALLOCATE" (cont.)

```
2070 REM ....................................................
2075 REM
2080 REM subroutine scanif -- second weapon
2085 J0=0
2090 L3=U0+R0
2095 L4=U0+1
2100 J=L3
2105 IF J<L4 OR J0<>0 THEN 2165
2110 T2=T(L,J)
2115 J1=INT(100*(T2/100-INT(T2/100)))
2120 IF W(J1)=0 THEN 2155
2125 J0=J
2130 G2=T2-INT(T2)
2135 X=G2-G2*G1
2140 Y=N1
2145 GOSUB 1705
2150 REM
2155 J=J-1
2160 GO TO 2105
2165 REM while-end (3320)
2170 RETURN
2175 REM ....................................................
2180 REM
2185 REM subroutine apply2 -- 2nd weapon
2190 W(J1)=W(J1)-M
2195 N1=N1-M
2200 A0=A0+1
2205 A(A0)=M*100+J1+K1/100
2210 IF F1 THEN 2230
2215 IF G0>G1+X THEN 2245
2220 D(C)=D(C)+M
2225 GO TO 2245
2230 D(C)=D(C)+M*X
2235 F3=0>D(C)
2240 REM
2245 N0=N0-M
2250 F3=N0>0 AND F3
2255 RETURN
```

"ALLOCATE" (cont.)

```
2260 REM ...................................................
2265 REM
2270 REM subroutine scanrt -- second weapon
2275 REM
2280 J0=0
2285 L0=U0+1
2290 L1=U0+R0
2295 J=L0
2300 IF J>L1 OR J0>0 THEN 2370
2305 T2=T(C,J)
2310 J1=INT(100*(T2/100-INT(T2/100)))
2315 G2=T2-INT(T2)
2320 X=G2-G1*G2
2325 IF G1+X<G0 THEN 2360
2330 IF W(J1)>0 THEN 2355
2335 J0=J
2340 Y=N1
2345 GOSUB 1705
2350 REM
2355 REM
2360 J=J+1
2365 GO TO 2300
2370 REM while-end (3725)
2375 IF J0>0 THEN 2385
2380 GOSUB 2080
2385 RETURN
2390 REM ...................................................
2395 REM subroutine report
2400 GO TO INT(G(C)/100) OF 2405,2415,2425,2435
2405 C$='cr-mide'
2410 GO TO 2440
2415 C$='cr-ide'
2420 GO TO 2440
2425 C$='tu-mide'
2430 GO TO 2440
2435 C$='tu-ide'
2440 REM
2445 PRINT C$;
2450 PRINT USING 2455:N(C);D(C);
2455 IMAGE 6d,-6d
2460 FOR J=1 TO W0-1
2465 PRINT USING 2470:T(C,J);
2470 IMAGE 7d.2d
2475 NEXT J
2480 PRINT USING 2470:T(C,W0)
2485 IF A0=0 THEN 2520
2490 PRINT USING 2495;
2495 IMAGE 20x
2500 FOR J=1 TO A0
2505 PRINT USING 2470:A(J);
2510 NEXT J
2515 PRINT
2520 RETURN
2525 REM ...................................................
3000 REM target for append
3005 REM target for append
3010 REM target for append
4000 REM target for append
4005 REM target for append
4010 REM target for append
```

Appendix B

IBM PERSONAL COMPUTER ANNOTATED LISTINGS

```
1000 '
1005 ' ALLOCATION program...................September 1982
1010 ' Prepared by Dr. Stefan Shrier
1015 '              System Planning Corporation
1020 '              1500 Wilson Blvd.
1025 '              Arlington, VA 22209
1030 '              Telephone (703)-841-3621
1035 '
1040 DIM T#(80,20),A#(20)
1045 DIM D(80),G(80)
1050 DIM W%(20),N%(80)
1055 DIM YTYPE$(4)
1060 PRINT "end declaration"
1065    GOSUB 1130                'get user's control toggles and wpn inventory
1070 FOR OBJ% = 1 TO NOBJ%
1075    GOSUB 1180                'get classes within this objective
1080     FOR CLASS% = 1 TO NCLASS%
1085            GOSUB 1220      'ALLOC1
1090            GOSUB 2285      'REPORT
1095            NEXT CLASS%
1100     FOR CLASS% = 1 TO NCLASS%
1105            GOSUB 1740      'ALLOC2
1110            GOSUB 2285      'REPORT
1115            NEXT CLASS%
1120    NEXT OBJ%
1125    END
1130 'subroutine to input toggles and wpn inventory
1135    INPUT "gtd-0 or dtd-1";A$
1140    INPUT "de goals cov-0 or mod-1 or hi-2"; H1%
1145    OPEN "inventor" FOR INPUT AS #1
1150    INPUT#1, U0%: FOR J%=1 TO U0%: INPUT#1,W%(J%): NEXT J%
1155    INPUT#1, B0%: FOR J%=1 TO B0%: INPUT#1,W%(J%+U0%): NEXT J%
1160    W0% = U0% + B0%:CLOSE#1
1165    OPEN "de-table" FOR INPUT AS #1
1170    INPUT#1, NOBJ%
1175    RETURN
1180 'subroutine to input DEs for classes within objective
1185    INPUT#1, NCLASS%
1190    FOR CLASS% = 1 TO NCLASS%
1195            INPUT#1, A$, N%(CLASS%),G(CLASS%)
1200            FOR J% = 1 TO W0%: INPUT#1, T#(CLASS%,J%): NEXT J%
1205            NEXT CLASS%
1210    RETURN
1215 '
1220 '
1225 'subroutine ALLOC1 -- allocate first wpn
1230    A0% = 0                   'clear second wpn pointer
1235    GOSUB 1290                'unpack the DEGOALs for this class
1240 '
1245    F3% = 1                   'flag to scan more
1250    WHILE F3%<>0              'meet the class requirement
1255            GOSUB 1390      'SCAN table for wpn to meet goal
1260            '
1265            IF J0% > 0 THEN GOSUB 1650 ELSE F3% = 0  'APPLY one else quit
1270            WEND
1275    RETURN
1280 '
1285 '
1290 '
1295 'subroutine DEGOAL -- unpack goals and categories for class
1300 '  categories (tu/crit and ide/mide) <--> flags F1 and F2
1305    G0 = INT (INT (G(CLASS%))/100!) 'highest digit marks category
1310    F1% = 0
1315    IF (G0 = 1) OR (G0 = 3) THEN F1% = 1    'f1 set for mide
1320    '
```

```
1325    F2% = 0
1330    IF G0 >= 3 THEN F2% = 1                      'f2 set for time urgent
1335    '
1340   G0 = G(CLASS%)/100 - G0
1345   'H1% = INT(10 * (G0 - INT(G0)))  'determine if coverage, moderate or hi
1350    IF H1%=2 THEN G0 = INT(G0*100)/100 ELSE G0 = G(CLASS%)-INT(G(CLASS%))
1355    '
1360    N0% = N%(CLASS%)    'number of targets in this class
1365    '                    D() either nr. tgts or mide goal
1370    IF F1% THEN D(CLASS%) = -N0% * G0 ELSE D(CLASS%) = -N0%
1375    RETURN
1380 '
1385 '
1390 '
1395 'subroutine SCAN(r) -- for first wpn
1400 '  scanning to the right in the DE table. J0 marks table column chosen
1405 '  select first wpn meeting per wpn DE goal
1410    J0% = 0: L0% = 1: L1% = U0%      'left half
1415    IF F2% = 0 THEN L0% = U0% + 1: L1% = U0% + B0%  'right half
1420    J% = L0%
1425    WHILE J%<= L1% AND J0% = 0
1430            T1# = T#(CLASS%,J%)
1435            G1 = T1# - INT(T1#)       'candidate wpn DE
1440            IF G1 < G0 THEN 1475     'wpn DE < per tgt DF goal
1445                    J1% = INT(T1#) MOD 100   'wpn index
1450                    IF W%(J1%) = 0 THEN 1470 'none left
1455                            J0% = J%         'Get table column index
1460                            X = G1: Y% = N0% 'arguments for HOWMNY subr.
1465                            GOSUB 1595       'HOWMANY wpns M
1470                            '
1475                            J% = J% + 1
1480            WEND
1485    IF J0% = 0 THEN GOSUB 1500        'if nothing found, try scanning left
1490    RETURN
1495 '
1500 'subroutine SCAN(l) for first wpn
1505    SWAP L0%,L1%
1510    J% = L0%
1515    WHILE J% > 0 AND J0% = 0
1520            T1# = T#(CLASS%,J%)
1525            J1% = INT(T1#) MOD 100            'wpn index
1535            IF W%(J1%) = 0 THEN 1565          'any wpns of this type left
1540                    J0% = J%
1545                    G1 = T1# - INT(T1#)       'wpn DE
1550                    X = G1: Y% = N0%          'arguments to subroutine
1555                    GOSUB 1595                'to compute HOWMNY wpns M
1560                    '
1565            J% = J% - 1
1570            WEND
1575            '
1580    RETURN
1585 '
1590 '
1595 '
1600 'subroutine HOWMNY -- args X (the wpn DE) and Y (wpn stock)
1605    IF F1% = 0 THEN 1625
1610            M% = INT (ABS(D(CLASS%))/X + 1)     'mide
1615            IF M% > Y% THEN M% = Y%
1620            GOTO 1630
1625    M% = Y%                                    'ide
1630    IF M% > W%(J1%) THEN M% = W%(J1%)    'more tgts than wpns?
```

```
1635    RETURN
1640 '
1645 '
1650 '
1655 'subroutine APPLY -- first wpn
1660    W%(J1%) = W%(J1%) - M%   'reduce inventory by nr. applied
1665    NO% = NO% - M%           'reduce tgts in class that get wpn
1670    T#(CLASS%,JO%) = T#(CLASS%,JO%) + 100 * M% 'record in table
1675    F3% = 1      '"more" := true. Set false below as appropriate
1680    IF NO% = 0 THEN F3% = 0          'false if no more tgts in class
1685    '
1690    IF F1% = 0 THEN 1715
1695            D(CLASS%) = D(CLASS%) + M% * X
1700            IF 0 <= D(CLASS%) THEN F3% = 0
1705            GOTO 1720
1710            '
1715    IF G0 < = G1 THEN D(CLASS%) = D(CLASS%) + M%
1720 '
1725    RETURN
1730 '
1735 '
1740 '
1745 'subroutine ALLOC2 -- shortfall to goal needs second wpn
1750    A0% = 0              'list pointer for 2nd wpns
1755    IF D(CLASS%) >= 0 THEN 1805      'return if no shortfall
1760            D1 = D(CLASS%)           'amount achieved so far
1765            GOSUB 1290               'DEGOALS for n0, g0, etc.
1770            '
1775            D(CLASS%) = D1           'get credit for earned DE
1780            '
1785            L0% = 1: L1% = U0%       'msl wpn
1790            IF F2% = 0 THEN L0% = U0% + 1: L1% = U0% + B0%
1795            GOSUB 1820       'execute ADD
1800            '
1805    RETURN
1810 '
1815 '
1820 '
1825 'subroutine ADD -- to try to achieve DE goals
1830    K% = L0%
1835    F3% = 1
1840    WHILE K% <= L1%
1845            T1# = T#(CLASS%,K%)
1850            G1 = T1# - INT(T1#)
1855            K0% = K%
1860               K1% = INT(100 * ((T1#/100) - INT(T1#/100)))
1865            IF F1% = 0 THEN F3% = G1 < G0 'if ide more if first wpn short
1870            N1% = INT(INT(T1#)/100)     'nr. wpns this type applied as first
1875            WHILE N1% > 0 AND F3%
1880                IF F1 THEN GOSUB 1920 ELSE GOSUB 2100 'mide lftscan else rt
1885                IF J0% > 0 THEN GOSUB 2005 ELSE F3% = 0
1890                WEND
1895            K% = K% + 1
1900            WEND
1905    RETURN
1910 '
1915 '
1920 'subroutine SCANLF -- second weapon
1925    J0% = 0
1930    L3% = U0% + B0%: L4% = U0% + 1
1935    J% = L3%
```

```
1940    WHILE J% >= L4% AND J0% = 0
1945            T2# = T#(CLASS%,J%)
1950            J1% = INT(T2#) MOD 100
1955            IF W%(J1%) = 0 THEN 1990
1960                    J0% = J%
1965                    G2 = T2# - INT(T2#)         '2nd wpn DE
1970                    X = G2 - G2 * G1            'marginal increase due to 2nd
1975                    Y% = N1%
1980                    GOSUB 1595       'HOWMANY of this 2nd?
1985                    '
1990            J% = J% - 1
1995            WEND
2000    RETURN
2005 '
2010 '
2015 'subroutine APPLY2 -- 2nd wpn
2020    W%(J1%) = W%(J1%) - M%   'reduce inventory by nr. applied
2025    N1% = N1% - M%
2030    A0% = A0% + 1    'advance list pointer for 2nd wpn data
2035    A#(A0%) = M% * 100 + J1% + K1%/100#
2040    IF F1% = 0 THEN 2065
2045            D(CLASS%) = D(CLASS%) + M% * X
2050            F3% = 0 > D(CLASS%)
2055            GOTO 2075
2060            '
2065    IF G0 <= G1 + X THEN D(CLASS%) = D(CLASS%) + M%
2070    '
2075    N0% = N0% - M%
2080    IF N0% = 0 THEN F3% = 0
2085    RETURN
2090 '
2095 '
2100 'subroutine SCANRT -- second weapon
2105 '
2110    J0% = 0
2115    L0% = U0% + 1: L1% = U0% + B0%
2120    J% = L0%
2125     WHILE J0% = 0 AND J% <= L1%
2130            T2# = T#(CLASS%,J%)
2135            J1% = INT(T2#) MOD 100
2140            G2 = T2# - INT(T2#)
2145            X = G2 - G2 * G1
2150            IF (G1 + X) < G0 THEN 2185
2155                    IF W%(J1%) > 0 THEN 2180
2160                            J0% = J%
2165                            Y% = N1%
2170                            GOSUB 1595       'HOWMNY wpns M?
2175                            '
2180                    '
2185            J% = J% + 1
2190            WEND
2195    IF J0% = 0 THEN 1920     'if none, try SCANLEFT
2200    RETURN
```

```
2285 'subroutine report
2290 '
2295    TTYPE$(1) = "cr-mide ":TTYPE$(2) = "cr-ide ":
2300    TTYPE$(3) = "tu-mide ":TTYPE$(4) = "tu-ide "
2305 '
2310    PRINT TTYPE$(INT(G(CLASS%)/100'));
2315    PRINT USING "######";N%(CLASS%);
2320    PRINT USING "######-";D(CLASS%);
2325    FOR J%=1 TO WO%-1: PRINT USING "#######.##";T#(CLASS%,J%);: NEXT J%
2330    PRINT USING "#######.##";T#(CLASS%,WO%)
2335    IF AO% = 0 THEN 2355
2340            PRINT "      ";"      ";"      ";"      ";
2345            FOR J%=1 TO AO%:PRINT USING "#######.##"; A#(J%);:NEXT J%
2350            PRINT
2355    '
2360    RETURN
2365    '
```

```
100 '
200 REM          adaptation of decalc.........august 1982
210 REM  Prepared by Dr. Stefan Shrier
220 REM  System Planning Corporation
230 REM  1500 Wilson Blvd.
240 REM  Arlington, VA 22209
250 REM  Telephone (703)-841-3621
310 OPEN "OUTFILE" FOR OUTPUT AS #5
400 DIM F(7,4),B(4,18),E(4,2),G(6,5),Q(2)
420 DIM D(4,20)
430 DIM W$(20) 'wpn IDs
460 REM
470 REM SET CONSTANT TABLES
480 REM
490 Q(1) = 1.04167
500 Q(2) = 1.0989
510 C$ = "pycv"
520 L$ = "lpmnorsqtu d"
740 FOR J = 1 TO 5
750     FOR I = 1 TO 6
760             READ G(I,J)
770             NEXT I
780     NEXT J
790 DATA 1.8622,- 3.237, 2.0771, -1.4128, 0.40348, -0.045266
800 DATA 1.813, -3.0865, 1.7386, -1.0635, 0.28357, -0.028466
810 DATA 1.8095, -2.858, 1.2935, -0.6232, 0.13978, -0.011672
820 DATA 1.7671, -2.636, 1.006, -0.35646, 0.06215, -0.004070752
830 DATA 1.6984, -2.3264, 0.74818, -0.18787, 0.023284, -0.0010853
831 REM
840 FOR J = 1 TO 4
850     FOR I = 1 TO 7
860             READ F(I,J)
870             NEXT I
880     NEXT J
890 DATA 8.214, -0.1118, 5.265 e -4, 2.162 e-5, -6.622 e-7, 7.132 e-9
900 DATA -3.064 e-11, 8.315, -0.1033, -7.908 e-4, -9.039 e-5, 1.458 e-5
910 DATA -5.22e-7, 5.726e-9, 8.783, -0.1355, 0.002355, -2.086e-4
920 DATA 9.961e-6, -1.872e-7, 1.227e-9, 8.789, -0.112, -6.658e-5
930 DATA -5.803e-4, 5.853e-5, -1.905e-6, 2.056e-8
931 REM
940 FOR I = 1 TO 4
950     FOR J = 1 TO 18
960             READ B(I,J)
970             NEXT J
980     NEXT I
990 DATA -0.5779, -1.[illegible]4, -1.957, -[illegible], -3.612, -5.051, -6.658, -8.926
1000 DATA -12.73, -0.[illegible]989, -0.6119, -0.[illegible], -1.401, -1.901, -2.509, -3.284
1010 DATA -4.[illegible], -6.[illegible], 1.569, 3.33, 5.[illegible], 7.742, 10.72, 14.63, 20.2, 28.98
1020 DATA 45, 0.7969, 1.729, 2.637, 4.1[illegible], 5.[illegible], 8.179, 11.32, 16.05, 24.43
10[illegible] [illegible] 0.00[illegible]376, -0.00[illegible]69, -0.04773, -0.[illegible], -0.8418, -2.375, -5.982
1040 DATA -14.12, 35.56, -0.03991, -0.1869, -0.5116, -1.133, -2.23, -4.116
1050 DATA [illegible].756, -13.21, -25.77, -0.00631, -0.04695, -0.1545, -0.7457
1060 DATA -0.5135, -0.4769, 0.2151, 3.11[illegible], 1[illegible].[illegible], 0.001152, 0.009542
10[illegible] DATA 0.0145, 0.1812, 0.4847, 1.12, 2.[illegible], 4.90[illegible], 10.64
1071 REM
10[illegible] FOR J = 1 TO 2
1090     READ F(1,J),E(2,J),F(7,J),E(4,J)
1100     NEXT J
11[illegible] DATA 7.171, -0.[illegible], -0.[illegible]1956, 2.2780 5
11[illegible] DATA 7.[illegible]4, -0.1574, 0.[illegible]278, -6.618 e-5
11[illegible] [illegible]
[illegible] [illegible] = .[illegible]   'CONSTANT FOR NORMAL DISTRIBUTION -- USED IN PD
[illegible] [illegible]      ****
    [illegible] get wpn dat
[illegible]
[illegible]     OPEN "arsenal" FOR INPUT AS #5
[illegible]          [illegible] ro, urgent,yield, cep, fa, yield, cep, fa, . . .
```

```
1710 INPUT#5,U0
1720 FOR J = 1 TO U0
1730     INPUT#5,W$(J),D(1,J),D(2,J),D(3,J)
1740     NEXT J
1750 REM
1760 REM same for non-u
1770 INPUT#5,B0
1780 FOR J = 1 TO B0
1790     J1 = J + U0
1800     INPUT#5,W$(J1),D(1,J1),D(2,J1),D(3,J1)
1810     NEXT J
1820 CLOSE#5
1830 W0 = U0 + B0
1900 REM
1910 REM     MAIN PROCESS LOOP ********
1920 OPEN "DE-TABLE" FOR OUTPUT AS #3
1930 OPEN "tgt-db" FOR INPUT AS #5
1940 REM
1950 INPUT#5,O1  'number of objectives
1955 PRINT#3,O1
1960 FOR O= 1 TO O1
1970     INPUT#5,C1    'number of classes
1975     PRINT#3, C1
1980     FOR C = 1 TO C1
1990             INPUT#5,T$,M,B$,I,N,G1,G2  'tgt id,type,ref,size,n ,hf gl
gl
2000             GOSUB 2330 'interpret vote input...
2010             GOSUB 2850 'compute DE
2015             GOSUB 2600 'prepare and write tables
2020             NEXT C
2030     NEXT O
2040 GOTO 4840 'end of program
2300 '
2310 '
2330 REM subroutine to interpret vote input
2340                     Z$ = MID$(B$,LEN(B$)-1,1)
2350                     P = INSTR(1,I$,Z$)
2360                     IF P = 0 THEN 2390
2370                         PRINT#6, "type must be l,a,n,o,p,q,r,s,t,u, ,1
2380                         RETURN
2390                     X$ = LEFT$(B$,LEN(B$)-2)
2400                     V = VAL(X$)
2410                     IF P 6 OR P=12 AND V 57 OR (P 11 AND V 35) THEN 2450
2420                     IF P=11 AND V 0 AND V =4 THEN 2450
2430                             PRINT#6, "vn out of range for tgt type"
2440                             RETURN
2450                     IF P 12 THEN 2470
2460                             P = 5
2470                     S = .1*P
2480                     T = P
2490                     IF P = 11 THEN 2550
2500                             IF P   6 THEN 2520
2510                                 S = S - .5
2520                             C$ = RIGHT$(B$,1)
2530                             P = VAL (C$)
2540                             RETURN
2550                     P = 0
2560                     RETURN
2600  'subroutine to prepare and write tables
2605   FOR J2 = 1 TO W0
2610          D(4,J2) = J2 + D(4,J2)
2615          NEXT J2
2620   L0 = 1
2625   L1 = U0
2630   GOSUB 4760 'sort by Pd
2635   L0 = U0 + 1
```

```
2640    L1 = U0 + B0
2645    GOSUB 4760
2650    PRINT#3, USING "\        \";T$;       'tgt-id 8-byte field
2655    PRINT#3, USING "\  \";B$;             'vntk
2660    PRINT#3, USING "#####";Z;            'size
2665    PRINT#3, USING "####";N;              'number
2670    PRINT#3, USING "####.##";M*100+INT(G1*100)+G2     'tgt-type & goals
2675    PRINT#3, "      ";                   '6-byte offset line two
2680    FOR J3 = 1 TO W0-1
2685            PRINT#3, USING "###.##";D(4,J3);          'wpn-index & Pd
2690            NEXT J3
2695    PRINT#3, USING "###.##";D(4,W0
2700    RETURN
2850 'subroutine to compute DEs
3060     H0 = 1 'air only assumption
3080     J1 = 1 'since only one target
3090     I4 = 1
3100     IF T<6 THEN 3120
3110     I4 = 2
3120     I8 = 10 * S 'needed in Pd subroutine
3130     FOR J3 = 1 TO W0
3150        Y3 = D(1,J3)^(1/3)
3160        IF T = 11 THEN 3300
3170        GOSUB 4360 'COMPUTE WPN-RADIUS
3180        GOSUB 4510 'OPT HOB
3300        'placeholder
3350        J4 = J3 'for same wpns
3360        J5 = 1 ' -- only one target size
3380        GOSUB 4580 'compute pd
3390        D(4,J3) = P
3950        Z$ = MID$(L$,T,1)
3960       ' PRINT#6, V;Z$;K;V1;D(1,J3);D(2,J4);Z;
3970       ' PRINT#6, H0;W;P;D(3,J3);D1
4300        NEXT J3
4310     RETURN
4320 '
4330 '
4360 'sub for radius
4361 Y4 = 1/Y3
4365     Y4 = 1/Y3
4370     V1 = V
4380     IF K = 0 THEN 4410
4390             J = 9*I4-9+K
4400             V1 = V + B(1,J) + Y4 * (B(2,J) + Y4 * (B(3,J) + Y4 * B(4,J)))
4410     J = I4 + 2
4440     W = F(7,J)
4450     FOR I = 6 TO 1 STEP -1
4460             W = W * V1 + F(I,J)
4470             NEXT I
4480     W = Y3 * EXP(W)
4490     W = W/(Q(I4) * (1 - S^2))
4500     RETURN
4510 'subroutine for opthob
4520    IF I4 = 1 OR V1 => 15 THEN 4550
4530            H0 = 1661 * EXP (-.06138 * V1)
4540            GOTO 4560
4550    H0 = EXP(E(1,I4) + V1*(E(2,I4) + V1*(E(3,I4) + V1 * E(4,I4))))
4560    H0 = Y3 * H0
4570 RETURN
4575 'subroutine for pd
4580     IF D(2,J4) THEN 4610 'if any cep is 0, set p := 1 and return
4590             P = 1
4600             RETURN
4610     R1 = W/SQR(D(2,J4)^2 + C7 * Z^2)
4620     IF S = .1 AND R1 > 3 THEN 4590
4630     IF S = .2 AND R1 >3.5 THEN 4590
4640     IF S =.3 AND R1>4.5 THEN 4590
```

```
4650      IF S =.4 AND R1>6 THEN 4590
4660      IF S =.5 AND R1>8 THEN 4590
4670      IF R1>.1 THEN 4710
4680              P = 0
4690              RETURN
4710      P9 = R1 * (G(5,I8) + R1 * G(6,I8))
4720      P9 = G(1,I8) + R1 * (G(2,I8) + R1 * (G(3,I8) + R1 * (G(4,I8) + P9)))
4730      P = EXP ( -EXP(P9))
4750      RETURN
4760 'subroutine to order in table
4765     IF L1 - LO < 1 THEN 4795
4770     FOR I = LO + 1 TO L1
4775             FOR J = LO TO I
4780                     IF (D(4,I)-INT(D(4,I)))<(D(4,J)-INT(D(4,J))) THEN
                            SWAP D(4,I),D(4,J)
4785                     NEXT J
4790             NEXT I
4795     RETURN
4840 'program is finished
4880 END
```

Appendix C

TEKTRONIX COMPUTER LISTINGS FOR PD CALC AND DE CALC (PROVIDED BY SAGA)

```
100 INIT
1000 DIM F(7,4),B(4,18),E(4,2),L$(11),B$(4),C$(1),O$(4),X$(2),Z$(1)
1010 DIM D(5,25),S(25),Z(5),P(6,50),C(4),U(5),T(25),Q(2)
1020 DIM G(6,5),R(5),P(16,8),H(4),V(25),L(25)
1030 IMAGE 3L,6T,"PAGE  ",3D,3L
1040 REM
1050 IMAGE 3D,1X,A,2D,2X,4D.1D,3X,6D.2X,6D,2X,6D,4X,6D,5X,6D.1D,4D.3D
1055 PAGE
1056 PRINT "DISPLAY OUTPUT AT (SCREEN=32, PRINTER=51)   "
1057 INPUT A
1058 PRINT "INITIALIZING PROBLEM ....."
1059 IF A=32 THEN 1060
1060 Y=20
1061 IF A=51 THEN 1062
1062 Y=40
1063 GO TO 1070
1064 PRINT "NO SUCH DEVICE"
1065 GO TO 1056
1070 C2=0
1075 D=0
1080 I7=0
1090 Q(1)=1.04167
1100 Q(2)=1.0989
1110 O$="PYCV"
1120 L$="LPMRORSUTUX"
1130 DATA 200,40,1,0,0.4,0.3,0.2,0,40,8,1.6,0,0.5,0.4,0.3,0
1140 DATA 81,0,0,0,0.3,0.2,0,0,71,31,1,0.1,0.4,0.4,0.3,0.3
1150 DATA 49,11,1,0,0.4,0.3,0.2,0,100,9,4,1.1,0.5,0.5,0.4,0.3
1160 DATA 31,0,0,0,0.2,0.3,0,0,101,9,4.1,1,0.4,0.4,0.3,0.3
1170 DATA 1375,1570,2420,0,0.37,0.34,0.21,0,2525,2220,2975,0,0.37,0.38
1180 DATA 0.26,0,1150,2675,0,0,0.33,0.17,0,0,2430,2500,2500,4135,0.36
1190 DATA 0.34,0.34,0.2,1600,1650,2650,0,0.39,0.39,0.18,0,4190,2900
1200 DATA 2540,3260,0.37,0.45,0.5,0,1375,2950,0,0,0.38,0.15,0,0,3750
1210 DATA 2715,2660,3450,0.37,0.44,0.43,0.24,800,1000,700,1000
1220 FOR I=1 TO 16
1230 FOR J=1 TO 8
1240 READ U(I,J)
1250 NEXT J
1260 NEXT I
1270 FOR I=1 TO 4
1280 READ H(I)
1290 NEXT I
1300 FOR I=1 TO 5
1310 READ R(I)
1320 NEXT I
1330 DATA 3,3.5,4,5,6,8
1340 FOR J=1 TO 5
1350 FOR I=1 TO 6
1360 READ G(I,J)
1370 NEXT I
1380 NEXT J
1390 DATA 1.8622,-3.2372,2.0272,-1.4129,0.49348,-0.095276
1400 DATA 1.843,-3.0855,1.7386,-1.0645,0.28357,-0.02966
1410 DATA 1.8025,-2.858,1.2935,-0.6732,0.13979,-0.01172
1420 DATA 1.7671,-2.635,1.095,-0.35846,0.04211,-0.000252
1430 DATA 1.6984,-2.3254,0.74918,-0.18784,0.02284,-0.001053
1440 FOR J=1 TO 4
1450 FOR I=1 TO 7
1460 READ F(I,J)
1470 NEXT I
1480 NEXT J
1490 DATA 8.214,-0.1118,5.265E-4,2.162E-5,-6.638E-7,7.132E-9
1500 DATA -3.064E-11,8.315,-0.1032,-7.908E-4,-9.039E-5,1.458E-5
1510 DATA -5.22E-7,5.726E-9,8.783,-0.1355,0.00255,-2.085E-4
1520 DATA 9.901E-6,-1.922E-7,1.229E-9,9.729,-0.1112,-2.808E-5
```

PREVIOUS PAGE IS BLANK

```
1525 DATA -5.803E-4,5.852E-5,-1.905E-6,2.056E-8
1530 FOR I=1 TO 4
1540 FOR J=1 TO 18
1550 READ B(I,J)
1560 NEXT J
1570 NEXT I
1580 DATA -0.5779,-1.224,-1.957,-2.805,-3.812,-5.051,-6.658,-8.926
1590 DATA -12.73,-0.2899,-0.6119,-0.9782,-1.401,-1.901,-2.509,-3.284
1600 DATA -4.343,-6.01,1.569,3.33,5.352,7.742,10.72,14.63,20.2,28.98
1610 DATA 45,0.7989,1.729,2.832,4.196,5.915,8.179,11.32,16.05,24.43
1620 DATA 0.001376,-0.002069,-0.01773,-0.2253,-0.8418,-2.375,-5.982
1630 DATA -14.22,-35.56,-0.03991,-0.1869,-0.5116,-1.133,-2.23,-4.116
1640 DATA -7.356,-13.21,-25.37,-0.00631,-0.01695,-0.1545,-0.3457
1650 DATA -0.5135,-0.4749,0.3151,3.115,12.22,0.001152,0.009542
1660 DATA 0.5145,0.1912,0.4847,1.12,2.376,4.907,10.64
1670 FOR J=1 TO 2
1680 READ E(1,J),E(2,J),E(3,J),E(4,J)
1690 NEXT J
1700 DATA 7.171,-0.0251,-0.001885,2.278E-5,7.154,-0.1576
1701 DATA 0.005228,-6.618E-5
1710 L3=1
1720 PRINT "DO YOU WANT A TAPE OUTPUT?"
1730 INPUT C$
1740 IF C$="N" THEN 1850
1760 IF C$="Y" THEN 1800
1780 PRINT "ANSWER MUST BE YES(Y) OR NO(N)
1790 GO TO 1720
1800 I7=1
1810 PRINT "FIRST FILE NUMBER (>4)    ";
1820 INPUT F1
1830 IF F1<5 THEN 1810
1840 GOSUB 4890
1850 L1=0
1860 L2=0
1870 PRINT "UNKNOWN:END,P,Y,C ? ";
1880 INPUT C$
1890 I1=POS(U$,C$,1)
1900 IF I1=0 THEN 4960
1910 I1=I1+1
1920 PRINT "TGT DIST (UNIFORM=0 NORMAL=1   ) ? ";
1930 INPUT C$
1940 IF C$="0" THEN 1970
1950 IF C$="1" THEN 1972
1960 GO TO 1920
1970 I2=0
1971 GO TO 1980
1972 I2=1
1980 PRINT "SINGLE PROBLEM INPUT ? ";
1990 INPUT C$
2000 I5=1
2010 I3=0
2030 IF C$="Y" THEN 2100
2060 IF C$="N" THEN 2110
2080 PRINT "ANSWER MUST YES(Y) OR NO(N) "
2090 GO TO 1972
2100 I3=1
2110 FOR I=1 TO 5
2120 IF I=I1 OR I3 THEN 2710
2130 GO TO I OF 2140,2160,2180,2200,2610
2140 PRINT "NUMBER OF UNIT'S ? ";
2150 GO TO 2640
2160 PRINT "NUMBER OF PD'S ? ";
2170 GO TO 2640
2180 PRINT "NUMBER OF YIELDS ? ";
2190 GO TO 2640
2200 IF I1=3 OR N(3)=1 THEN 2330
```

```
2210 PRINT 'CEP'S VARY INDEPENDENTLY ? ';
2220 INPUT C$
2240 IF C$='Y' THEN 2330
2260 IF C$='N' THEN 2300
2280 PRINT 'ANSWER MUST BE YES(Y) OR NO(N) '
2290 GO TO 2210
2300 N(4)=R(3)
2310 I5=0
2320 GO TO 2730
2330 PRINT 'CEP IN INCREMENTS ? ';
2340 INPUT C$
2350 IF C$='N' THEN 2630
2370 IF C$='Y' THEN 2410
2390 PRINT 'ANSWER MUST BE YES(Y) OR NO(N) '
2400 GO TO 2330
2410 PRINT 'CEP(MIN), CEP(MAX),STEP ? ';
2420 INPUT C1,C2,C3
2430 IF C2-C1>0 THEN 2460
2440 PRINT 'CEP(MAX) MUST BE >CEP(MIN)'
2450 GO TO 2410
2460 IF C3<C2-C1 THEN 2490
2470 PRINT 'STEP MUST BE <CMAX-CMIN'
2480 GO TO 2410
2490 IF C1=>0 AND C2>0 AND C3>0 THEN 2520
2500 PRINT 'CEP MUST BE >0'
2510 GO TO 2410
2520 J2=(C2-C1)/C3+1
2530 IF J2<=25 AND J2>=1 THEN 2560
2540 PRINT '25 IS THE MAX NUMBER OF CEPS'
2550 GO TO 2410
2560 FOR J1=1 TO J2
2570 D(4,J1)=C1+(J1-1)*C3
2580 NEXT J1
2590 N(I)=J2
2600 GO TO 2720
2610 PRINT 'NUMBER OF TGT RADII ? ';
2620 GO TO 2640
2630 PRINT 'NUMBER OF CEP'S ? ';
2640 INPUT N(I)
2650 IF N(I)<=25 AND N(I)>=1 THEN 2730
2660 PRINT 'NUMBER MUST BE >=1 AND <=25'
2670 GO TO I OF 2140,2160,2180,2630,2610
2680 FOR J9=1 TO 4
2690 N(J9)=1
2700 NEXT J9
2710 N(I)=1
2720 IF I=I1 THEN 3210
2730 FOR J=1 TO N(I)
2740 GO TO I OF 2750,3000,3020,3040,3070
2750 PRINT 'VR(K('J;')= ';
2760 INPUT B$
2762 C$=SEG(B$,LEN(B$)-1,1)
2770 Z$=C$
2780 P=POS(L$,C$,1)
2790 IF P THEN 2830
2810 PRINT 'TYPE MUST BE L,M,N,O,P,Q,R,S,T,U,X'
2820 GO TO 2750
2830 X$=SEG(B$,1,LEN(B$)-2)
2835 V(J)=VAL(X$)
2840 IF P<6 AND V(J)<57 OR (P<11 AND V(J)<35) THEN 2880
2850 IF P=11 AND V(J)>0 AND V(J)<=4 THEN 2890
2860 PRINT 'VR IS OUT OF RANGE FOR TGT TYPE'
2870 GO TO 2750
2880 S(J)=0.1*P
2890 T(J)=P
2900 IF P=11 THEN 2980
```

```
2910 IF P<6 THEN 2940
2920 S(J)=S(J)-0.5
2940 C$=SEG(B$,LEN(B$),1)
2945 K(J)=VAL(C$)
2950 GO TO 3200
2980 K(J)=0
2990 GO TO 3200
3000 PRINT 'FD('$J$')=  ';
3010 GO TO 3080
3020 PRINT 'YIELD-KT('$J$')= ';
3030 GO TO 3080
3040 IF C2<>0 THEN 3210
3050 PRINT 'CEP-FT('$J$')= ';
3060 GO TO 3080
3070 PRINT 'TGT RADIUS-FT('$J$')= ';
3080 INPUT D(I,J)
3090 IF I<>2 OR (D(I,J)<0.999 AND D(I,J)>0.02) THEN 3120
3100 PRINT 'FD MUST BE <.999 AND >.02'
3110 GO TO 3000
3120 IF I<>3 OR D(I,J)>=0 THEN 3150
3130 PRINT 'YIELD MUST BE >0'
3140 GO TO 3020
3150 IF I<>4 OR D(I,J)>=0 THEN 3170
3160 GO TO 3050
3170 IF I<>5 OR D(I,J)>=0 THEN 3200
3180 PRINT 'TGT RADIUS MUST BE >=0'
3190 GO TO 3070
3200 NEXT J
3210 NEXT I
3220 IF I5 THEN 3240
3230 N(4)=1
3240 PRINT 'HOB(GND 0,AIR=1,BOTH=2) ? ';
3250 INPUT I6
3255 REM SET TRACE
3260 IF I6=0 OR I6=1 OR I6=2 THEN 3290
3270 PRINT 'HOB MUST=0;1;2'
3280 GO TO 3240
3290 N(6)=1
3300 IF I6<>2 THEN 3320
3310 N(6)=2
3320 FOR J6=1 TO N(6)
3330 H0=0
3340 IF I6=0 OR (I6=2 AND J6=1) THEN 3360
3350 H0=1
3360 FOR J1=1 TO N(1)
3370 I4=1
3380 IF T(J1)<6 THEN 3400
3390 I4=2
3400 FOR J2=1 TO N(2)
3410 FOR J3=1 TO N(3)
3420 IF I1=3 THEN 3590
3430 Y3=D(3,J3)^(1/3)
3440 IF T(J1)=11 THEN 3580
3450 GOSUB 4510
3460 GOSUB 4660
3470 GO TO 3580
3480 I=T(J1)
3490 V1=V(J1)
3500 IF H0=0 THEN 3530
3510 I=I+1
3520 H0=H(V1)*Y3
3530 FOR J=1 TO 4
3540 IF D(3,J3)<=U(I,J) THEN 3560
3550 NEXT J
3560 [illegible]U(I+8,J)*D(3,J3)^U(I+8,J+1)
3570 [illegible](J1)=U(I,J+1)
```

```
3580 FOR J4=1 TO N(4)
3590 C7=0.4
3600 IF I2=0 THEN 3620
3610 C7=0.231
3620 IF I5 THEN 3640
3630 J4=J3
3640 FOR J5=1 TO N(5)
3650 IF I1<>2 THEN 3680
3660 GOSUB 4730
3670 GO TO 4070
3680 F=D(2,J2)
3690 B2=LOG(-LOG(F))
3700 I8=10*S(J1)
3710 F8=(B2-G(1,I8))/G(2,I8)
3720 K1=G(3,I8)/G(2,I8)
3730 K2=G(4,I8)/G(2,I8)
3740 K3=G(5,I8)/G(2,I8)
3750 K4=G(6,I8)/G(2,I8)
3760 U1=0
3770 U2=R(I8)
3780 U3=(U1+U2)/2
3790 F7=U3*(1+U3*(K1+U3*(K2+U3*(K3+U3*K4))))
3800 IF ABS(F7-F8)<=1.0E-3 THEN 3860
3810 IF F7-F8>0 THEN 3840
3820 U1=U3
3830 GO TO 3780
3840 U2=U3
3850 GO TO 3780
3860 IF I1=3 THEN 3900
3870 D(4,1)=W/U3
3880 D(4,1)=D(4,1)^2-C7*D(5,J5)^2
3881 IF D(4,1)>0 THEN 3885
3882 D(4,1)=-1
3883 GO TO 3890
3885 D(4,1)=SQR(D(4,1))
3890 GO TO 4070
3900 W1=U3
3910 W1=SQR(D(4,J4)^2+D(5,J5)^2*C7)*W1
3920 M1=0
3930 M2=10
3940 Y3=M2
3950 GOSUB 4510
3960 IF W>W1 THEN 4000
3970 M1=M2
3980 M2=2*M2
3990 GO TO 3940
4000 Y3=(M1+M2)/2
4010 GOSUB 4510
4020 IF W>W1 THEN 4036
4030 M1=Y3
4035 GO TO 4040
4036 M2=Y3
4040 IF ABS(W-W1)/W1>1.0E-3 THEN 4000
4050 D(3,1)=Y3^3
4060 GOSUB 4650
4070 REM
4080 IF L1 THEN 4125
4090 L2=L2+1
4095 PAGE
4100 PRINT @A: USING 1030:L2
4102 PRINT @A:" VN  I.I.     AVH     YIELD(KT) CEP(FT) IRAD(FT)";
4105 PRINT @A:"      HOB(FT)    WR(FT)      FD"
4110 PRINT @A:
4120 REM                              IF D(3,J3)>100 THEN 4130
```

```
4125 Z$=SEG(L$,T(J1),1)
4130 PRI #6: USI 1050:V(J1),Z$,K(J1);V1;D(3,J3);D(4,J4),D(5,J5),H0,W,P
4140 GO TO 4160
4160 L1=L1+1
4170 IF I7=0 THEN 4420
4180 IF T(J1)=10 THEN 4220
4190 IF T(J1)=11 THEN 4240
4200 T9=T(J1)
4210 GO TO 4270
4220 T9=0
4230 GO TO 4270
4240 T9=1
4250 U(1,L3)=-(V(J1)+T9/10+K(J1)/100)*100
4260 GO TO 4280
4270 U(1,L3)=(V(J1)+T9/10+K(J1)/100)*100
4280 U(2,L3)=D(3,J3)
4290 U(3,L3)=D(4,J4)
4300 U(5,L3)=D(5,J5)
4310 U(4,L3)=0
4320 IF H0=0 THEN 4340
4330 U(4,L3)=1
4340 U(6,L3)=1000+P
4350 L3=L3+1
4360 IF L3<51 THEN 4420
4370 REM STORE DATA
4375 STOP
4380 F1=F1+1
4390 GOSUB 4890
4400 L3=1
4410 IF L1<Y THEN 4440
4420 IF L1<Y THEN 4440
4430 L1=0
4440 NEXT J5
4450 NEXT J4
4460 NEXT J3
4470 NEXT J2
4480 NEXT J1
4490 NEXT J6
4500 GO TO 1850
4510 Y4=1/Y3
4520 V1=V(J1)
4530 IF K(J1)=0 THEN 4560
4540 J=9*I4-9+K(J1)
4550 V1=V(J1)+B(1,J)+Y4*(B(2,J)+Y4*(B(3,J)+Y4*B(4,J)))
4560 J=I4
4570 IF H0=0 THEN 4590
4580 J=J+2
4590 W=F(7,J)
4600 FOR I=6 TO 1 STEP -1
4610 W=W*V1+F(I,J)
4620 NEXT I
4630 W=Y3*EXP(W)
4640 W=W/(Q(I4)*(1-S(J1)^2))
4650 RETURN
4660 IF H0=0 THEN 4750
4670 IF I4=1 OR V1=>15 THEN 4700
4680 H0=1661*EXP(-0.06139*V1)
4690 GO TO 4710
4700 H0=EXP(E(1,I4)+V1*(E(2,I4)+V1*(E(3,I4)+V1*E(4,I4))))
4710 H0=Y3*H0
4720 RETURN
4730 IF D(4,J4) THEN 4760
4740 P=1
4750 RETURN
4760 R1=W/SQR(D(4,J4)^2+C7*D(5,J5)^2)
4770 IF S(J1)=0.1 AND R1<3 THEN 4740
```

```
4780 IF S(J1)=0.2 AND R1>3.5 THEN 4740
4790 IF S(J1)=0.3 AND R1>4.5 THEN 4740
4800 IF S(J1)=0.4 AND R1>6 THEN 4740
4810 IF S(J1)=0.5 AND R1>8 THEN 4740
4820 IF R1>0.1 THEN 4850
4830 P=0
4840 RETURN
4850 I8=10*S(J1)
4860 P9=R1*(G(5,I8)+R1*G(6,I8))
4865 P9=G(1,I8)+R1*(G(2,I8)+R1*(G(3,I8)+R1*(G(4,I8)+P9)))
4870 P=EXP(-EXP(P9))
4880 RETURN
4890 FOR J=1 TO 50
4900 FOR I=1 TO 6
4910 U(I,J)=0
4920 NEXT I
4930 NEXT J
4935 RETURN
4940 FIND F1
4950 RETURN
4960 IF I7=0 OR L3=1 THEN 4980
4970 REM STORE DATA
4980 IF I7=0 THEN 5000
4990 REM LOAD DATA
5000 PRINT 'PROGRAM IS FINISHED'
5001 END
```

```
100 INIT
110 PAGE
120 PRINT ' DDDDD    EEEEEEE        CCCC      A      L          CCCC  '
130 PRINT ' D    D   E             C    C    A A     L         C    C '
140 PRINT ' D     D  E            C         A   A    L        C       '
150 PRINT ' D     D  E            C        A     A   L        C       '
160 PRINT ' D     D  EEEEE        C        AAAAAAA   L        C       '
170 PRINT ' D     D  E            C        A     A   L        C       '
180 PRINT ' D     D  E            C        A     A   L        C       '
190 PRINT ' D    D   E        OO   C    C  A     A   L         C    C '
200 PRINT ' DDDDD    EEEEEEE  OO    CCCC   A     A   LLLLLLL    CCCC  '
210 PRINT ' '
220 PRINT '*************************************************************'
230 PRINT
240 REM   EDITED  BY:  MAJ BOB STUART
250 REM                FTD/SDBS
260 REM                WRIGHT-PATTERSON AFB, OH 45433
270 REM                513-257-2542
280 REM                AV  787-2542
290 REM
300 REM   REVISED ON:  20 APR 82
310 REM
320 REM
330 REM
340 REM
350 REM
360 PRINT 'INITIALIZING THE PROBLEM ...'
370 PRINT
380 PRINT
390 PRINT
400 DIM F(7,4),B(4,19),E(4,2),L$(12),B$(4),C$(1),U$(4),X$(2),Z$(1)
410 DIM D(6,50),S(50),Z(6),O(8,50),C(4),P(7),T(50),Q(2),T$(12),W$(20)
420 DIM G(6,5),R(5),U(16,8),H(4),V(50),K(50),S$(4)
430 IMAGE 34T,12A,4X,20A/6T,'PAGE  ',3D,3L
431 IMAGE 2L,34T,12A,4X,20A
440 IMAGE 3D,1X,A,2D,6D.1D,1X,3(9D),S
450 IMAGE 10D,8D.1D,3(5D.3D)
460 C2=0
470 D=0
480 I7=0
485 L3=1
490 Q(1)=1.04167
500 Q(2)=1.0989
510 U$='PYCU'
520 L$='LFMNORSUTUXH'
521 S$='UCST'
523 REM
525 FOR I=1 TO 16
526 FOR J=1 TO 8
527 READ U(I,J)
528 NEXT J
529 NEXT I
530 DATA 200,40,1,0,0.4,0.3,0.2,0
535 DATA 40,8,1.6,0,0.5,0.4,0.3,0
540 DATA 81,0,0,0,0.3,0.3,0,0
545 DATA 71,31,5,0.1,0.4,0.4,0.3,0.3
550 DATA 40,11,1,0,0.4,0.3,0.2,0
555 DATA 100,9,4,1.1,0.5,0.5,0.4,0.3
560 DATA 31,0,0,0,0.3,0.3,0,0
565 DATA 101,9,4.1,1,0.4,0.4,0.3,0.2
570 DATA 1375,1570,2420,0,0.37,0.31,0.21,0
575 DATA 2525,2360,2975,0,0.37,0.39,0.26,0
580 DATA 1150,2675,0,0,0.36,0.17,0,0
585 DATA 2430,2500,2500,3135,0.36,0.35,0.34,0.2
590 DATA 1680,1650,2650,0,0.39,0.39,0.18,0
595 DATA 4190,2900,2510,3240,0.37,0.45,0.5,0
```

```
600 DATA 1375,2950,0,0,0.38,0.15,0,0
610 DATA 3750,2715,2660,3450,0.37,0.44,0.43,0.24
611 REM
670 FOR I=1 TO 4
680 READ H(I)
690 NEXT I
695 DATA 800,1000,700,1000
696 REM
700 FOR I=1 TO 5
710 READ R(I)
720 NEXT I
730 DATA 3,3.5,4.5,6,8
731 REM
740 FOR J=1 TO 5
750 FOR I=1 TO 6
760 READ G(I,J)
770 NEXT I
780 NEXT J
790 DATA 1.8622,-3.237,2.0771,-1.4128,0.40348,-0.045265
800 DATA 1.943,-3.0865,1.7386,-1.0625,0.28257,-0.028466
810 DATA 1.8095,-2.859,1.2935,-0.4232,0.13728,-0.011672
820 DATA 1.7671,-2.636,1.006,-0.35546,0.06215,-0.0040252
830 DATA 1.6984,-2.3264,0.74818,-0.18783,0.022284,-0.0030853
831 REM
840 FOR J=1 TO 4
850 FOR I=1 TO 7
860 READ F(I,J)
870 NEXT I
890 NEXT J
890 DATA 8.214,-0.1118,5.265E-4,2.162E-5,-6.638E-7,7.132E-9
900 DATA -3.064E-11,8.315,-0.1033,-7.908E-4,-9.039E-5,1.458E-5
910 DATA -5.22E-7,5.726E-9,8.783,-0.1355,0.002355,-2.086E-4
920 DATA 9.901E-6,-1.822E-7,1.227E-9,8.789,-0.112,-6.658E-5
930 DATA -5.803E-4,5.853E-5,-1.905E-6,2.056E-8
931 REM
940 FOR I=1 TO 4
950 FOR J=1 TO 18
960 READ B(I,J)
970 NEXT J
980 NEXT I
990 DATA -0.5779,-1.224,-1.957,-2.805,-3.812,-5.051,-6.658,-8.926
1000 DATA -12.73,-0.2889,-0.6119,-0.9782,-1.401,-1.901,-2.509,-3.284
1010 DATA -4.343,-6.01,1.569,3.33,5.357,7.742,10.72,14.63,20.2,28.98
1020 DATA 45,0.7989,1.729,2.837,4.196,5.915,8.179,11.32,16.05,24.43
1030 DATA 0.001376,-0.002069,-0.04773,-0.2253,-0.8418,-2.375,-5.982
1040 DATA -14.22,-35.56,-0.03991,-0.1869,-0.5116,-1.133,-2.23,-4.116
1050 DATA -7.356,-13.21,-25.37,-0.00631,-0.04695,-0.1545,-0.3457
1060 DATA -0.5135,-0.4769,0.3151,3.115,12.22,0.001152,0.009542
1070 DATA 0.5145,0.1812,0.4847,1.12,2.376,4.907,10.64
1071 REM
1080 FOR J=1 TO 2
1090 READ E(1,J),E(2,J),E(3,J),E(4,J)
1100 NEXT J
1110 DATA 7.171,-0.0251,-0.00186,2.278E-5
1115 DATA 7.154,-0.1576,0.005228,-6.618E-5
1120 REM
1121 PRINT "OUTPUT IS: (U)NCLASS, (C)ONFID, (S)ECR, OR (T)OP SEC ? ";
1122 INPUT C$
1123 GO TO POS(S$,C$,1) OF 1124,1126,1128,1130
1124 T$="UNCLASSIFIED"
1125 GO TO 1137
1126 T$="CONFIDENTIAL"
1127 GO TO 1137
1128 T$="    SECRET    "
1129 GO TO 1137
1130 T$=" TOP SECRET"
```

```
1131 GO TO 1137
1132 PRINT "SORRY!"
1133 PRINT "THIS TERMINAL IS NOT CLEARED FOR ",C$
1134 PRINT "NEITHER ARE YOU!!!"
1135 PRINT "PLEASE TRY AGAIN."
1136 GO TO 1121
1137 PRINT "ENTER CODEWORD OR OTHER CAVEAT ";
1138 INPUT W$
1139 REM
1140 PRINT "DISPLAY OUTPUT AT (SCREEN=32, PRINTER=51)    ";
1150 INPUT A
1160 IF A=32 THEN 1200
1170 IF A=51 THEN 1220
1180 PRINT "NO SUCH DEVICE"
1190 GO TO 1140
1200 Y=20
1210 GO TO 1280
1220 Y=50
1230 PRINT "SELECT THIS MACHINE ON PRINTER SWITCH"
1240 PRINT "PRINTER LINE SWITCH - ON"
1250 PRINT "ADJUST PRINTER TO TOP OF PAGE"
1260 PRINT "TYPE <CR> WHEN READY"
1270 INPUT C$
1271 REM
1280 PRINT "DO YOU WANT A TAPE OUTPUT?";
1290 INPUT C$
1300 IF C$="N" THEN 1390
1310 IF C$="Y" THEN 1340
1320 PRINT "ANSWER MUST BE YES(Y) OR NO(N)   "
1330 GO TO 1280
1340 I7=1
1350 PRINT "FIRST FILE NUMBER (>4)    ";
1360 INPUT F1
1370 IF F1<5 THEN 1350
1380 GOSUB 4760
1390 L1=-1
1400 L2=0
1410 PAGE
1411 REM
1420 PRINT "UNKNOWN=END;P,Y,C ? ";
1430 INPUT C$
1440 I1=POS(D$,C$,1)
1450 IF I1=0 THEN 4840
1460 I1=I1+1
1461 REM
1470 PRINT "TGT DIST (UNIFORM=0 ;NORMAL=1   ) ? ";
1480 INPUT C$
1490 IF C$="0" THEN 1520
1500 IF C$="1" THEN 1540
1510 GO TO 1470
1520 I2=0
1530 GO TO 1550
1540 I2=1
1541 REM
1550 PRINT "SINGLE PROBLEM INPUT ? ";
1560 INPUT C$
1570 I5=1
1580 I3=0
1590 IF C$="Y" THEN 1630
1600 IF C$="N" THEN 1640
1610 PRINT "ANSWER MUST YES(Y) OR NO(N)
1620 GO TO 1550
1630 I3=1
1635 REM LOOP ON I
1640 FOR I=1 TO 6
1650 IF I=11 OR 13 THEN 2270
```

```
1660 GO TO I OF 1670,1690,1710,1730,2150,2170
1661 REM
1670 PRINT 'NUMBER OF VRTK'S ? ';
1680 GO TO 2200
1681 REM
1690 PRINT 'NUMBER OF PD'S ? ';
1700 GO TO 2200
1701 REM
1710 PRINT 'NUMBER OF YIELDS ? ';
1720 GO TO 2200
1730 IF I1=3 OR N(3)=1 THEN 1830
1731 REM
1740 PRINT 'CEP'S VARY INDEPENDENTLY ? ';
1750 INPUT C$
1760 IF C$='Y' THEN 1830
1770 IF C$='N' THEN 1800
1780 PRINT 'ANSWER MUST BE YES(Y) OR NO(N) '
1790 GO TO 1740
1800 N(4)=N(3)
1810 I5=0
1820 GO TO 2290
1821 REM
1830 PRINT 'CEP IN INCREMENTS ? ';
1840 INPUT C$
1850 IF C$='N' THEN 2190
1860 IF C$='Y' THEN 1890
1870 PRINT 'ANSWER MUST BE YES(Y) OR NO(N) '
1880 GO TO 1830
1890 PRINT 'CEP(MIN), CEP(MAX),STEP ? ';
1900 INPUT C1,C2,C3
1910 IF C2-C1>0 THEN 1940
1920 PRINT 'CEP(MAX) MUST BE >CEP(MIN)'
1930 GO TO 1890
1940 IF C3<C2-C1 THEN 1970
1950 PRINT 'STEP MUST BE <CMAX-CMIN'
1960 GO TO 1890
1970 IF C1=>0 AND C2>0 AND C3>0 THEN 2000
1980 PRINT 'NEED CEP >= 0'
1990 GO TO 1890
2000 J2=(C2-C1)/C3+1
2020 IF J2<=50 AND J2=>1 THEN 2100
2030 PRINT 'YOU HAVE REQUESTED';J2;' CEP'S'
2040 PRINT 'DO YOU WANT THAT MANY (Y/N)?'
2050 INPUT C$
2060 IF C$='Y' THEN 2100
2070 IF C$='N' THEN 1890
2080 GO TO 2050
2085 REM     LOOP ON J1
2100 FOR J1=1 TO J2
2110 D(4,J1)=C1+(J1-1)*C3
2120 NEXT J1
2125 REM     END OF   J1
2130 N(I)=J2
2140 GO TO 2280
2141 REM
2150 PRINT 'NUMBER OF TGT RADII ? ';
2160 GO TO 2200
2170 N(6)=N(3)
2180 GO TO 2290
2181 REM
2190 PRINT 'NUMBER OF CEP'S ? ';
2191 REM
2192 REM
2200 INPUT N(I)
2210 IF N(I)<=50 AND N(I)>=1 THEN 2290
2220 PRINT 'NUMBER MUST BE >=1 AND <=50'
```

```
2230 GO TO I OF 1670,1690,1710,2190,2150,2170
2231 REM
2235 REM   LOOP ON J9
2240 FOR J9=1 TO 4
2250 N(J9)=1
2260 NEXT J9
2265 REM     END OF   J9
2270 N(I)=1
2280 IF I=11 THEN 2840
2289 REM     LOOP ON   J
2290 FOR J=1 TO N(I)
2300 GO TO I OF 2310,2570,2590,2610,2640,2650
2301 REM
2310 PRINT "VNTK(";J;")= ";
2320 INPUT B$
2330 Z$=SEG(B$,LEN(B$)-1,1)
2350 P=POS(L$,Z$,1)
2360 IF P THEN 2390
2370 PRINT "TYPE MUST BE L,M,N,O,P,Q,R,S,T,U,X,D"
2380 GO TO 2310
2390 X$=SEG(B$,1,LEN(B$)-2)
2400 V(J)=VAL(X$)
2410 IF P<6 OR P=12 AND V(J)<57 OR (P<11 AND V(J)<35) THEN 2450
2420 IF P=11 AND V(J)>0 AND V(J)<=4 THEN 2470
2430 PRINT "VN IS OUT OF RANGE FOR TGT TYPE"
2440 GO TO 2310
2450 IF P<12 THEN 2470
2460 P=5
2470 S(J)=0.1*P
2480 T(J)=P
2490 IF P=11 THEN 2550
2500 IF P<6 THEN 2520
2510 S(J)=S(J)-0.5
2520 C$=SEG(B$,LEN(B$),1)
2530 K(J)=VAL(C$)
2540 GO TO 2830
2550 K(J)=0
2560 GO TO 2830
2561 REM
2570 PRINT "FD(";J;")=  ";
2580 GO TO 2670
2581 REM
2590 PRINT "YIELD-KT(";J;")= ";
2600 GO TO 2670
2610 IF C2<>0 THEN 2840
2611 REM
2620 PRINT "CEP-FT(";J;")= ";
2630 GO TO 2670
2631 REM
2640 PRINT "TGT RADIUS-FT(";J;")= ";
2650 GO TO 2670
2651 REM
2660 PRINT "WSR (";J;")= ";
2670 REM
2671 REM
2675 INPUT D(I,J)
2676 REM
2680 IF I<>2 OR (D(I,J)<0.999 AND D(I,J)>0.02) THEN 2710
2690 PRINT "FD MUST BE <.999 AND >.02"
2700 GO TO 2570
2705 REM
2710 IF I<>3 OR D(I,J)>0 THEN 2740
2720 PRINT "YIELD MUST BE >0"
2730 GO TO 2590
2735 REM
2740 IF I<>4 OR D(I,J)>=0 THEN 2770
```

```
2750 PRINT "CEP MUST BE >=0"
2760 GO TO 2620
2765 REM
2770 IF I<>5 OR D(I,J)>=0 THEN 2800
2780 PRINT "TGT RADIUS MUST BE >=0"
2790 GO TO 2640
2795 REM
2800 IF I<>6 OR (D(I,J)<=1 AND D(I,J)>0.02) THEN 2830
2810 PRINT "WSR MUST BE <=1.0 AND >.02"
2820 GO TO 2660
2830 NEXT J
2835 REM     END OF   J
2840 NEXT I
2845 REM  END OF   I
2850 IF I5 THEN 2870
2860 N(4)=1
2865 REM
2870 PRINT "HOB(GND=0,AIR=1,BOTH=2) ? ";
2880 INPUT I6
2890 REM SET TRACE
2900 IF I6=0 OR I6=1 OR I6=2 THEN 3000
2910 PRINT "HOB MUST=0,1,2"
2920 GO TO 2870
2930 REM
2940 REM END OF INPUT ROUTINE
2950 REM
2960 REM ************************************************************
2970 REM
2980 REM START COMPUTATIONS
2990 REM
3000 N(7)=1
3010 IF I6<>2 THEN 3030
3020 N(7)=2
3025 REM  LOOP ON J7
3030 FOR J7=1 TO N(7)
3040 H0=0
3050 IF I6=0 OR (I6=2 AND J7=1) THEN 3080
3060 H0=1
3065 REM     LOOP ON J1
3080 FOR J1=1 TO N(1)
3090 I4=1
3100 IF T(J1)<6 THEN 3120
3110 I4=2
3115 REM       LOOP ON J2
3120 FOR J2=1 TO N(2)
3125 REM         LOOP ON J3
3130 FOR J3=1 TO N(3)
3140 IF I1=3 THEN 3300
3150 Y3=D(3,J3)^(1/3)
3160 IF T(J1)=11 THEN 3300
3170 GOSUB 4360
3180 GOSUB 4510
3190 GO TO 3300
3200 I=V(J1)
3210 V1=V(J1)
3220 IF H0=0 THEN 3250
3230 I=I+4
3240 H0=H(V1)*Y3
3245 REM           LOOP ON J
3250 FOR J=1 TO 4
3260 IF D(3,J3)<=U(I,J) THEN 3280
3270 NEXT J
3275 REM           END OF   J
3280 W=U(I+9,J)*D(3,J3)^U(I+8,J+4)
3290 S(J1)=U(I,J+4)
3295 REM           LOOP ON J4
```

```
3300 FOR J4=1 TO N(4)
3310 C7=0.4
3320 IF I2=0 THEN 3340
3330 C7=0.231
3340 IF I5 THEN 3360
3350 J4=J3
3355 REM     LOOP ON J5
3360 FOR J5=1 TO N(5)
3370 IF I1<>2 THEN 3410
3380 GOSUB 4580
3390 D1=P*D(6,J3)
3400 GO TO 3860
3410 P=D(2,J2)
3420 B2=LOG(-LOG(P))
3430 I8=10*S(J1)
3440 P8=(B2-G(1,I8))/G(2,I8)
3450 K1=G(3,I8)/G(2,I8)
3460 K2=G(4,I8)/G(2,I8)
3470 K3=G(5,I8)/G(2,I8)
3480 K4=G(6,I8)/G(2,I8)
3490 U1=0
3500 U2=R(I8)
3510 U3=(U1+U2)/2
3520 P7=U3*(1+U3*(K1+U3*(K2+U3*(K3+U3*K4))))
3530 IF ABS(P7-P8)<=1.0E-3 THEN 3590
3540 IF P7-P8>0 THEN 3570
3550 U1=U3
3560 GO TO 3510
3570 U2=U3
3580 GO TO 3510
3590 IF I1=3 THEN 3670
3610 D(4,1)=(W/U3)^2-C7*D(5,J5)^2
3620 IF D(4,1)>0 THEN 3650
3630 D(4,1)=-1
3640 GO TO 3660
3650 D(4,1)=SQR(D(4,1))
3660 GO TO 3860
3680 W1=SQR(D(4,J4)^2+D(5,J5)^2*C7)*U3
3690 M1=0
3700 M2=10
3710 Y3=M2
3720 GOSUB 4360
3730 IF W>W1 THEN 3770
3740 M1=M2
3750 M2=2*M2
3760 GO TO 3710
3770 Y3=(M1+M2)/2
3780 GOSUB 4360
3790 IF W>W1 THEN 3820
3800 M1=Y3
3810 GO TO 3830
3820 M2=Y3
3830 IF ABS(W-W1)/W1>1.0E-3 THEN 3770
3840 D(3,1)=Y3^3
3850 GOSUB 4510
3860 REM
3870 IF L1>0 THEN 3950
3875 PRINT L1
3876 L2=L2+1
3879 IF L1=0 THEN 3885
3880 L1=0
3882 PRINT L1
3884 GO TO 3900
3885 PRINT @A: USING 431:I$,W$
3890 PRINT @A:"L"
3900 PRINT @A: USING 430:I$,N$,L2
```

```
3910 PRINT @A:' VR T K     AVN     YIELD(KT) CEP(FT) TRAD(FT)';
3920 PRINT @A:'    HOB(FT)    WR(FT)        FD    X    PA    -    DE'
3930 PRINT @A:
3940 REM           WHAT's THIS??? ===>>>           IF D(3,J3)<100 THEN 4130
3950 Z$=SEG(L$,T(J1),1)
3960 PRINT @A: USING 440:V(J1),Z$,K(J1),V1,D(3,J3),D(4,J4),D(5,J5)
3970 PRINT @A: USING 450:H0,W,P,D(6,J3),D1
3990 L1=L1+1
4000 IF I7=0 THEN 4250
4010 IF T(J1)=10 THEN 4050
4020 IF T(J1)=11 THEN 4070
4030 T9=T(J1)
4040 GO TO 4100
4050 T9=0
4060 GO TO 4100
4070 T9=1
4080 U(1,L3)=-(V(J1)+T9/10+K(J1)/100)*100
4090 GO TO 4110
4100 U(1,L3)=(V(J1)+T9/10+K(J1)/100)*100
4110 U(2,L3)=D(3,J3)
4120 U(3,L3)=D(4,J4)
4130 U(5,L3)=D(5,J5)
4131 U(7,L3)=D(6,J3)
4132 U(8,L3)=D1
4140 U(4,L3)=0
4150 IF H0=0 THEN 4170
4160 U(4,L3)=1
4170 U(6,L3)=1000+P
4180 L3=L3+1
4190 IF L3<51 THEN 4250
4200 REM STORE DATA
4201 L3=1
4202 GO TO 4250
4210 STOP
4220 F1=F1+1
4230 GOSUB 4760
4240 L3=1
4250 IF L1<Y THEN 4280
4270 L1=0
4280 NEXT J5
4290 NEXT J4
4300 NEXT J3
4310 NEXT J2
4320 NEXT J1
4340 NEXT J7
4341 PRINT
4342 PRINT
4343 PRINT 'DO YOU HAVE MORE INPUTS (Y/N) ? ';
4344 INPUT C$
4345 IF C$='Y' THEN 1390
4346 GO TO 4840
4348 REM
4360 REM SUBROUTINE TO COMPUTE WEAPON RADIUS
4361 Y4=1/Y3
4370 V1=V(J1)
4380 IF K(J1)=0 THEN 4410
4390 J=9*I4-9+K(J1)
4400 V1=V(J1)+B(1,J)+Y4*(B(2,J)+Y4*(B(3,J)+Y4*B(4,J)))
4410 J=I4
4420 IF H0=0 THEN 4440
4430 J=J+2
4440 W=F(7,J)
4450 FOR I=6 TO 1 STEP -1
4460 W=W*V1+F(I,J)
4470 NEXT I
4480 W=Y3*EXP(W)
```

```
4490 W=W/(U(14)*(1-S(J1)^2))
4500 RETURN
4505 REM
4510 REM SUBROUTINE TO COMPUTE OPTIMUM HOB
4511 IF HO=0 THEN 4570
4520 IF J4=1 OR V1=>15 THEN 4550
4530 HO=1661*EXP(-0.06138*V1)
4540 GO TO 4560
4550 HO=EXP(E(1,14)+V1*(E(2,14)+V1*(E(3,14)+V1*E(4,14))))
4560 HO=Y3*HO
4570 RETURN
4575 REM
4576 REM SUBROUTINE TO
4580 IF D(4,J4) THEN 4610
4590 P=1
4600 RETURN
4610 R1=W/SQR(D(4,J4)^2+C7*D(5,J5)^2)
4620 IF S(J1)=0.1 AND R1>3 THEN 4590
4630 IF S(J1)=0.2 AND R1>3.5 THEN 4590
4640 IF S(J1)=0.3 AND R1>4.5 THEN 4590
4650 IF S(J1)=0.4 AND R1>6 THEN 4590
4660 IF S(J1)=0.5 AND R1>8 THEN 4590
4670 IF R1>0.1 THEN 4700
4680 P=0
4690 RETURN
4700 I8=10*S(J1)
4710 F9=R1*(G(5,I8)+R1*G(6,I8))
4720 F9=G(1,I8)+R1*(G(2,I8)+R1*(G(3,I8)+R1*(G(4,I8)+F9)))
4730 P=EXP(-EXP(F9))
4750 RETURN
4755 REM
4756 REM  SUBROUTINE TO STORE OUTPUT ON TAPE
4760 FOR J=1 TO 50
4770 FOR I=1 TO 6
4780 D(I,J)=0
4790 NEXT I
4800 NEXT J
4810 RETURN
4815 REM
4816 REM  SUBROUTINE TO
4820 FIND F1
4830 RETURN
4840 IF L1=0 THEN 4845
4841 IF L1<0 THEN 4847
4842 FOR I=L1 TO Y
4843 PRINT @A:
4844 NEXT I
4845 PRINT @A: USING 431:T$,W$
4846 PRINT @A:"L"
4847 IF I7=0 OR L3=1 THEN 4860
4850 REM STORE DATA
4860 IF I7=0 THEN 4870
4870 PRINT "PROGRAM IS FINISHED"
4880 END
```

Appendix D

SAMPLE ALLOCATION PROCEDURE
(PROVIDED BY SAGA)

ALLOCATION PROCEDURE

1. Computations

a. Installations within the target data base fall into one of two categories of DE goals. Installation Damage Expectancy (IDE) implies that each installation must be damaged to the required level. Mean Installation Damage Expectancy (MIDE) implies that installation set must be damaged to the required level. When there is a MIDE goal, the DE on some installations within the set may be greater than the goal while other installations may be damaged less than the goal and some may not be damaged at all.

b. Damage Expectancy (DE) is the product of the Probability of Destruction (PD) and the Probability of Arrival (PA). DE = PD x PA

c. The DE on an installation attacked by two weapons each with a damage expectancy of DE_1 and DE_2 is called compound damage expectancy and is expressed $CDE = 1-(1-DE_1) \times (1-DE_2)$

d. For sets of targets with an IDE goal the above relationship holds for each installation. Therefore, if it required "w" weapons of a given type to achieve the DE required on a single installation within the set, the total number of weapons that would be required to attack the set is (Nr of installations) x (w). Note that "w" must be an integer.

e. For sets with an MIDE goal, the following criteria are used:

(1) When the weapon DE against an installation within the set is greater than the MIDE goal the following relationship holds:

(MIDE Goal) x (Nr of Installations) = (Nr of Wpns) x (Wpn DE)

$$\text{Therefore, Nr of Wpns} = \frac{\text{(MIDE Goal) x (Nr of Instal)}}{\text{Wpn DE}}$$

For example, suppose the DE goal on a set of 100 installations is 0.5 and the weapon DE = 0.8. The number of weapons required is

$$n = \frac{(0.5) \times (100)}{0.8} = 62.5 = 63 \text{ weapons}$$

(2) When the weapon DE against an installation is less than the MIDE goal the following relationship holds:

$$\text{MIDE Goal} = 1 - (1-DE_1)^p \times (1 - DE_2)^q$$

where "p" and "q" are the numbers of weapons #1 and #2 applied, respectively against a single installation. Note that "p" and "q" do not need to be integers.

(a) Assuming one weapon #1 was applied against each installation in the set, the number "q" is the number of #2 weapons that need to be applied per installation. Solving for q,

$$\text{MIDE Goal} = 1 - (1 - DE_1)^1 \times (1 - DE_2)^q$$

$$(1 - DE_1) \times (1 - DE_2)^q = 1 - \text{MIDE Goal}$$

$$(1 - DE_2)^q = \frac{1 - (\text{MIDE Goal})}{1 - DE_1}$$

$$q \log (1 - DE_2) = \log \frac{1 - \text{MIDE Goal}}{1 - DE_1}$$

$$q = \frac{\log \dfrac{1 - \text{MIDE Goal}}{1 - DE_1}}{\log (1 - DE_2)}$$

The number of #2 weapons that must be allocated against the set, N, is

$$N = (q) \times (\text{Nr of installations})$$

(b) If only one type of weapon is used,

$$\text{MIDE Goal} = 1 - (1 - DE)^n$$

$$n = \frac{\log (1 - \text{MIDE Goal})}{\log (1 - DE)}$$

(c) If an equal number of two different weapons are to be used

$$\text{MIDE Goal} = 1 - (1 - DE_1)^n \ (1 - DE_2)^n$$

and

$$n = \frac{\log \ 1 - \text{MIDE Goal}}{\log \ (1 - DE_1)(1 - DE_2)}$$

$$N - (n) \times (\text{Nr of installations})$$

(3) The process described in paragraph (2) above will yield a number slightly less than the actual requirement.

(a) For example, suppose the MIDE goal on a set of 1000 installations is 0.7 and the weapon being used has a DE of .48. Using the formula from para (2)(b):

$$n = \frac{\log(1 - .7)}{\log(1 - .48)} = 1.841$$

N = (1.841) (1000) = 1841 weapons allocated against the set

Therefore, 841 installations are targeted with two weapons with DE = $1 - (1 - .48)^2 = .73$, and 195 installations are targeted with one weapon with DE = .48.

$$841 \times .73 = 613.9$$

$$159 \times .48 = \underline{\ 76.3\ }$$

$$690.2$$

The MIDE is .69 which is less than the goal of 0.7.

(b) A linear correction can be applied in the following manner. Let x be the additional number of installations to receive a second weapon

$$(841 + x)\ (.73) + (159 - x)\ (.48) = 700$$

$$690.25 + .25x = 700$$

x = 39 additional installations

Therefore, 1880 weapons must be allocated. The MIDE achieved is:

$$880 \times .73 = 642.4$$

$$120 \times .48 = \underline{\ 57.6\ }$$

$$700. \Rightarrow \text{MIDE} = .70$$

(c) During the allocations in this study, the linear correction was not computed. Instead, the individual performing the allocation was satisfied to achieve a DE within one percent of the goal.

c. Suppose that the PD against the majority of the installation in the target base was very close to 1.0. The DE achieved would be driven down by PA only. The impact of this is that the planner who established the DE goal may have envisioned severe damage to 70% of the target (for an area target) as the goal, not a 70% chance of a weapon arriving. If more than one weapon is targeted against an installation the probability of at least one weapon arriving is

$$P_{\substack{\text{at least one}\\ \text{arriving}}} = 1 - \prod_{i=1}^{n} (1 - PA_i)$$

where PA_i is the probability the i^{th} weapon will arrive.

2. Allocation Process.

a. The process by which weapons were allocated against the target base was the same in the generated and day-to-day cases with one exception. In the day-to-day case, when the DE goal was a goal against a set of installations (MIDE), weapons were allocated until the goal was reached even if all installations were not planned to be struck. In the generated case all installations were assigned weapons even if DE goals were exceeded.

b. In general, weapons were first allocated to achieve high confidence DE goals against the entire target base. If the high goal requirements could not be achieved the weapons were reallocated to achieve the moderate confidence DE goals. If the moderate goals could not be achieved, one weapon was assigned to each installation; and the DE was computed.

c. Time-urgent installations were allocated a missile warhead. ICBMs were struck using ICBM weapons. However, when the DE for an ICBM weapon against a silo was less than the DE for an SLBM weapon, the SLBM weapon was chosen.

d. Weapons were allocated against target objectives in a sequential order. Within each target objective, target classes identified as requiring time-urgent weapons were assigned a weapon first. Within the time-urgent and non-time-urgent categories, weapons were assigned first to sets of installations in which each installation was required to be damaged to the required level (IDE target classes) then to the sets in which the set was required to be damaged (MIDE target classes).

e. For each target class the DE associated with each available weapon system was computed. The weapon used was the one which most closely satisfied the DE requirement. However, if the DE of a weapon was more than 10 percent below the DE goal (e.g., weapon DE .45 against a DE goal of .50), the weapon was not used unless it was the most effective weapon remaining in the arsenal. Time-urgent targets were allocated missile weapons even if other weapon DE's were closer to the goal.

f. Generally, if the DE achieved was within 10 percent of the goal (e.g., above .45 against a .50 goal) the goal was considered to be met. There were a few exceptions. For example, if the IDE goal was .90 and the only weapons available h'd a DE of .80, one weapon was allocated instead of allocating a second weapon that would achieve a compound DE of .96.

g. Weapons were allocated against one target objective at a time using the above procedure. Each target class within the target objective was assigned weapons even if goals were not met. Before proceeding to the next target objective a second weapon was assigned to those target classes for which goals had not been reached. After all installations within the entire target set had been assigned weapons, a third weapon was assigned to installations where the DE goal had not been achieved.

h. Since the allocation was by target objective, time-urgent targets in some target objectives were assigned weapons after non-time urgent targets in high priority target objectives. Therefore, the individual performing the allocation "saved" missile weapons for those time urgent installations as he proceeded through the allocation process.

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

US Documents Officer
AFSOUTH
ATTN: US Doc Officer

Armed Forces Radiobiology Rsch Institute
ATTN: Dir

Armed Forces Staff College
ATTN: Library

Assistant Secretary of Defense
International Security Affairs
ATTN: Policy Plans & NSC Affairs

Assistant to the Secretary of Defense
Atomic Energy
ATTN: Mil Appl, COL Kahn
ATTN: Executive Assistant

Commander-in-Chief, Pacific
ATTN: J-3

Defense Intelligence Agency
ATTN: DT, Sci-Tecn Intell
ATTN: DB
ATTN: DE, Estimates
ATTN: DB-6, Rsch, Tgt Intell Div, MAJ Mann
5 cy ATTN: DB-4, Rsch, Resources Div

Defense Nuclear Agency
ATTN: NATA
ATTN: STNA
ATTN: STRA
ATTN: RAEE
ATTN: NATD
ATTN: NAFD
ATTN: STSP
4 cy ATTN: TITL

Defense Tech Info Ctr
12 cy DD

Deputy Under Secretary of Defense, S&TNF
ATTN: T. Jones

Field Command Defense Nuclear Agency
Det 1, Lawrence Livermore Lab
ATTN: FC-1

DOD SALT Task Force
ATTN: Dir Nuclear Policy

Field Command
Defense Nuclear Agency
2 cy ATTN: FCPR

Joint Chiefs of Staff
ATTN: SAGA-SFD
ATTN: SAGA
ATTN: J-5, Strat Div, W. McClain
ATTN: SAGA/SSD
ATTN: GD10, J-5, Nuc & Chem Div
ATTN: ED30, J-3, Strategic Ops Div
ATTN: J-3
3 cy ATTN: J-5, Nuc Div/Strat Div/FP&P Div

DEPARTMENT OF DEFENSE (Continued)

Joint Strat Tgt Planning Staff
ATTN: JL, Nat Strat Tgt List Dir
ATTN: JPTM
ATTN: JP, SIOP Dir
ATTN: JPPF
ATTN: JLA, Threat Applications Div

National Defense University
ATTN: NWCLB-CR
ATTN: ICAF, Tech Library

Office of the Secretary of Defense
Net Assessments
ATTN: F. Giessler
ATTN: Dir, A. Marshall
ATTN: C. Pease

Program Analysis & Evaluation
ATTN: Naval Forces
2 cy ATTN: Strategic Programs
2 cy ATTN: Regional Programs

US European Command
ATTN: ECJ-LW
ATTN: ECJ2-T, Tgts Div
ATTN: ECJ-2
ATTN: ECJ-2-ITD
ATTN: ECJ-3
ATTN: ECJ-5
ATTN: ECJ-5N, Nuc Div
ATTN: ECJ-5NPG
ATTN: ECJ-6

US National Military Representative
SHAPE
ATTN: US Docs Officer

Under Secretary of Defense for Policy
ATTN: USD/P
ATTN: DUSP/P

Under Secretary of Defense for Rsch & Engrg
ATTN: Strategic & Space Sys, (OS)
ATTN: Tactical Warfare Programs
ATTN: Strat & Theater Nuc Forces, B. Stephan
ATTN: Chairman, Def Science Brd

DEPARTMENT OF THE ARMY

Deputy Chief of Staff for Ops & Plans
ATTN: DAMO-NC, Nuc Chem Dir
ATTN: DAMO-ZXA
ATTN: DAMO-NCN
ATTN: DAMO-RQS

Harry Diamond Labs
ATTN: DELHD-TA-L, 81100

US Army Air Defense School
ATTN: Commandant

US Army Comd General Staff College
ATTN: ACQ, Library Div

DEPARTMENT OF THE ARMY (Continued)

US Army Concepts Analysis Agency
ATTN: CSSA-ADL

US Army Nuclear & Chemical Agency
ATTN: Library

US Army Training and Doctrine Comd
ATTN: ATCD-N, Cbt Dev, Nuc Dir

US Army War College
ATTN: Library

USA Military Academy
ATTN: Doc Library

USA Missile Command
ATTN: DRSMI-YDR

DEPARTMENT OF THE NAVY

Naval Intelligence Support Ctr
ATTN: NISC-30

Naval Ocean Systems Ctr
ATTN: Code 4471

Naval Postgraduate School
ATTN: Code 1424, Library

Naval Research Lab
ATTN: Code 2627, Tech Library

Naval Sea Systems Cmd
ATTN: SEA-09G53

Naval War College
ATTN: Code E-11, Tech Service

Ofc of the Deputy Chief of Naval OPs
ATTN: NOP 50, Avn Plns & Rqmts Dev
ATTN: NOP 654, Strat Eval & Anal Br

Office of Naval Rsch
ATTN: Tech Info Svcs

Office of the Chief of Naval Operations
ATTN: OP-00K

DEPARTMENT OF THE AIR FORCE

Air Force
ATTN: INA

Air Force
ATTN: INT

Headquarters
Air Force Systems Command
ATTN: SD
ATTN: XR
ATTN: DL

Air Force Weapons Lab
ATTN: SUL

Air University Library
ATTN: AUL-LSE

DEPARTMENT OF THE AIR FORCE (Continued)

Air War College
ATTN: EDRX

Assistant Chief of Staff
Studies & Analyses
ATTN: AF/SASB, H. Adams
ATTN: AF/SASF, Strat Frc Anal, Frc Anal
ATTN: AF/SAMI, W. Mattson
2 cy ATTN: AF/SASM
2 cy ATTN: AF/SAMI, Tech Info Div

Assistant Chief of the Air Force
Research, Dev & Logistics
ATTN: SAF/ALR

Deputy Chief of Staff
Research, Dev & Acq
ATTN: AFXDQI

Deputy Chief of Staff
Plans and Operations
ATTN: AFXOOIR
ATTN: AFXOXFS, Frc Dev, Strat Off Frc
ATTN: AFXOXFM, Plns, Frc Dev Mun Plns

Strategic Air Command
ATTN: XP
ATTN: DO
ATTN: INT
ATTN: XO
ATTN: XOXO
ATTN: STIC, 544 SIW
ATTN: XPFS, Maj Skluzacek
ATTN: NR
ATTN: NRI-STINFO, Library
ATTN: ADWA
ATTN: AD

US Air Force
ATTN: SASF, Lt Col Starr

DEPARTMENT OF ENERGY

Department of Energy
Albuquerque Operations Office
ATTN: CTID

Department of Energy
Office of Military Application
ATTN: OMA, DP-22

OTHER GOVERNMENT AGENCIES

Central Intelligence Agency
ATTN: NIO, Strategic Sys

Federal Emergency Management Agency
ATTN: Ofc of Rsch/NP, [illegible] Bensen
ATTN: Asst Assoc Dir for Rsch, J. Kerr
ATTN: G. Orrell, NP-CP

US Arms Control & Disarmament Agcy
ATTN: A. Lieberman

DEPARTMENT OF ENERGY CONTRACTORS

Los Alamos National Lab
ATTN: Reports Library
ATTN: M/S634, T. Dowler

DEPARTMENT OF ENERGY CONTRACTORS (Continued)

University of California
Lawrence Livermore National Lab
ATTN: Tech Info Dept, Library

Sandia National Labs
ATTN: 5513, R. Stratton
ATTN: Tech Library, 3141

DEPARTMENT OF DEFENSE CONTRACTORS

Advanced International Studies Institute
ATTN: M. Harvey

Advanced Research & Applications Corp
ATTN: Doc Control

Aerospace Corp
ATTN: Library

Analytical Assessments Corp
ATTN: A. Wagner

BDM Corp
ATTN: J. Conant
ATTN: J. Bode
ATTN: R. Buchanan
ATTN: C. Wasaff
ATTN: J. Braddock

Boeing Aerospace Co
ATTN: MS-85-20, D. Choate
ATTN: MS-85-20, J. Russel

Caci, Inc - Federal
ATTN: A. Berry

66th MI Group
Attention R & D Associates
ATTN: K. Moran

Computer Sciences Corp
ATTN: F. Eisenbarth

Horizons Techology, Inc
ATTN: J. Palmer

Hudson Institute, Inc
ATTN: H. Kahn

IIT Research Institute
ATTN: Doc Library

Institute for Defense Analyses
ATTN: Classified Library
ATTN: J. Grote

IRT Corp
ATTN: W. Macklin

JAYCOR
ATTN: R. Sullivan

Kaman Sciences Corp
ATTN: F. Shelton

Kaman Sciences Corp
ATTN: E. Daugs

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Kaman Tempo
ATTN: DASIAC

Kaman Tempo
ATTN: DASIAC

Martin Marietta Corp
ATTN: F. Marion

Martin Marietta Denver Aerospace
ATTN: J. Donathan

Orion Research Inc
ATTN: J. Scholz

Pacific-Sierra Research Corp
ATTN: H. Brode, Chairman SAGE
ATTN: G. Lang

Pacific-Sierra Research Corp
ATTN: D. Gormley

R & D Associates
ATTN: F. Field
ATTN: E. Carson
ATTN: W. Graham
ATTN: P. Haas
2 cy ATTN: Doc Con

R & D Associates
ATTN: J. Thompson

Rand Corp
ATTN: P. Davis
ATTN: V. Jackson

Rand Corp
ATTN. B. Bennett

Rockwell International Corp
ATTN: J. Howe, 113-129 MB 51

S-CUBED
ATTN: K. Pyatt

Santa Fe Corp
ATTN: D. Paolucci

Science Applications, Inc
ATTN: E. Swick
ATTN: J. Beyster
ATTN: Doc Con
ATTN: M. Drake
ATTN: J. Warner
ATTN: J. Martin

Science Applications, Inc
ATTN: Doc Con
ATTN: W. Layson
ATTN: J. Foster
ATTN: L. Goure
ATTN: J. Shannon

Science Applications, Inc
ATTN: D. Kaul

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Science Applications, Inc
ATTN: R. Craver

SRI International
ATTN: R. Tidwell

SRI International
ATTN: C Hulbert

System Planning Corp
ATTN: J. Ballentine
4 cy ATTN: S. Shrier

Systems Research & Applications Corp
ATTN: S. Greenstein

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

T. N. Dupuy Assoc, Inc
ATTN: T. Dupuy

Tetra Tech, Inc
ATTN: F. Bothwell

Titan Systems, Inc
ATTN: R. Lee

TRW Electronics & Defense Sector
ATTN: D. Skally
ATTN: R. Burnett
ATTN: N. Lipner

TRW Electronics & Defense Sector
ATTN: P. Dai